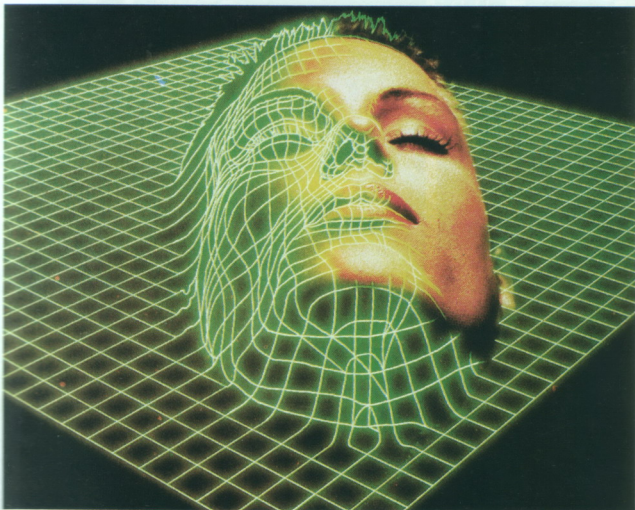


specijalno izdanje  
časopisa „Galaksija“  
mart 1987.  
cena 400 din.

izdaje BIGZ

izlazi jedanput mesečno

# računari 24



razglednica iz njujorka

**jedna  
američka  
priča**

komercijalni softver

**laser  
genius**

programeri govore  
džef raskin  
konstruktor računara  
„mekintoš“

**„mrzim  
miševe!“**

računarski algoritmi

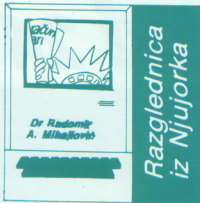
**kako  
nacrtati krug**

organizacija računara

**ibm pc bios**

umetak na 32 strane

**sa bejzika  
na paskal**



Razglednica  
iz Njujorka

## Jedna američka priča

**Kao kreator prvog portabl mikroručnara i osnivač kompanije sa najbržim usponom koji je silicijska dolina ikada videla, Adam Osborn je definitivno uzueo značajno mesto u istoriji elektronskih računara. Kako i na koji način je Osborn postao legenda? Na ovo pitanje je moguće odgovoriti tipičnom američkom pričom. Pričom o mu-njevitom usponu, još bržem padu i ponovnom uzdizanju „iz pepela“.**



Ponovo u akciji: Osborn sa suprugom Barbarom Bardik

puta, vlasnik najpoznatijeg lanca robnih kuća u Americi J. C. Peni je više puta finansijski padoo na kolena. Nolan Busnel, osnivač poznatog Atarija, propao je u projektu da uspostavi 1000 picerija za decu sa kompjuterizovanim igračkama u njima, Džin Amdal (Gene Amdahl), otac proslavljenog džambo-računara IBM-360 je nedavno priznao poraz u pokušaju da kreira novi računarski super-čip...

Mada Amerikanci duboko poštuju junake tipa Roki Balboa, koji iz poraza grade pobeđu, generalni stav prema onima koji gube je negativan. Puritanski pogled na uspeh i poraz je tesno vezan za moralne kvalitete pojedinca. Poslovni poraz baca sumnje ne samo na poslovne sposobnosti pojedinca, već i na karakter i ličnost „nesrećnika“. Sve do početka ovog veka zakon o bankrotstvu u Americi nije pružao nikakvu zaštitu onome koji je finansijski propadao. Kreditori su imali apsolutnu moć nad dužnicima. Nemilosrdno uzajamno uništavanje u borbi za opstanak bilo je osnovno etičko pravilo kako među kaubojima i indijancima Divljeg zapada, tako i među biznismenima „civilizovanog“ i industrijalizovanog Istoka. „Bog mi je dao milion dolara“, uzvikivao je Džon D. Rokfeleer, sa ubeđenjem da samo najizdržljiviji i najposposobniji treba da uspeju. U toku tridesetih godina, u vreme velikih svetskih ekonomskih kriza, ljudi bez posla i propali biznismeni su „popularno“ nazivani „mrtvacima“. U situaciji gde svaki drugo novo preduzeće propada u toku prve četiri godine postojanja, tradicionalni pogledi na poslovni brodolom se u uslovima modernog agilnog američkog kapitalizma polako me-

nja. Priča o Adamu Osbornu je dokaz takvog trenda.

## Leteći Osborn

Osborn je postigao svoj prvi veći poslovni poen na sajmu mikroručnara u San Francisku 1961. godine. U to vreme industrija mikroručnara je bila još uvek u povoju. Mnogi od štandova su bili sklepani od kartona, dok je veliki broj izloženih računara bio napravljen u nečijoj garaži. Osbornov stil je bio potpuno suprotan stidljivim proizvođačima skromnih „mikrača“ na sajmu. Njegov je štand bio najveći i najluksuzniji, sa velikim natpisom na vrhu „Leteći O“. Veliki natpis je kao magnet privlačio posetioce, tako da je gužva oko Osbornovog portabla neprekidno trajala. Daleka 1980. je bila godina velikih poslovnih avantura. Svakog dana, verovali ili ne, u Americi je startovalo novih 2.200 preduzeća, više nego što se rodi ljudi u Jugoslaviji. U to vreme je Epi bio tek tri godine star, dok je IBM smatrao industrijom mikroručnara neobjavljnom i beznačajnom. Osbornova Kompjuterska Korporacija je bila jedna od 100 kompanija koje su proizvele mikroručnare. Osborn se na samom početku izdvojio od ostalih opredeljenjem da proizvodi jeftino i minijaturno. Hteo je da izgradi računaru ekvivalentan po popularnosti „folksvagenu“ ili, pak, Fordovom „modelu-T“.

Karizmatičan i originalan, Osborn je privlačio veliku pažnju novinara, javnog mnjenja, i što je najvažnije, finansijera. U

Četnaestog novembra 1982. godine u popularnoj emisiji „60 minuta“ vodeće američke TV mreže C.B.S. Osborn je izjavio da će lično „najuriti“ i zadržati pisacu mašinu iz svih kancelarija na svetu. Prevedio je da će svaki od njegovih golobradih menadžera postati milioner pre tridesete. Deset meseci kasnije, na nesumnjivo veliku žalost kreditora, Adam Osborn je podneo zahtev za likvidaciju kompanije „Osborn Kompjuter“. Sa ukupnim gubicima od preko 45 miliona dolara, finansijski bljesak, za kratko najsjajnija zvezda, nestaje sa neba silicijske doline. Adam Osborn uzdignute glave izjavljuje: „Svako u igri avanturističkog investiranja u nove ideje mora da bude svestan da je potpuni gubitak uvek moguć“.

## Pobeda iz poraza

Finansijski krah je neminovni detalj u sveukupnoj slici agresivnog američkog kapitalizma. Henri Ford je bankrotirao dva

vrlu kratkom periodu je uspeo da sakupi skoro milion dolara za svoj istorijski poduhvat — prvi svetski portabl računar, težak 12 kilograma, koji se lako može smestiti ispod sedišta aviona, sa popularnom cenom od svega 1.795 dolara, više od dva puta nižom od cene ekvivalentnog „epi“ računara.

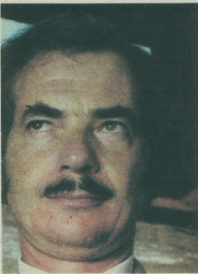
Danas već 45-to godišnjak, Adam Osborn je proveo svoje najranije detinjstvo sa svojim roditeljima, Britancima, u Indiji. Ma da su mu roditelji bili posvećeni jednoj od indijskih religija, Osborn je stekao svoje prvo obrazovanje u striktno organizovanoj Katoličkoj školi. „Još kao dečak sam je samoinicijativno konvertovao u katolika. Dovoljan razlog za takvu odluku bila mi je činjenica da su časne sestre u školi boje tretirale dečake katolike od nekatolika. Opravdano je sve što je korisno, priseća se Osborn.

## Računarski meteor

Doktorsku disertaciju u oblasti hemijskog inženjerstva Osborn je odbranio na Univerzitetu Delaver u Sjedinjenim Državama. Po završetku postdiplomskih studija zapošljava se u Selu, odakle ga, po njegovim rečima, posle nekoliko godina izbacuju. Započinje sa novinarskim radom. Postaje stalni dopisnik tada vrlo popularnog magazina „Doba Interfejsa“ (Interface Age). Glavna teza mu je bila da lideri rasteuće industrije mikroročunara nisu vizionari da su bezbedni i kratkovični. Pošto mu je podelila knjiga o mikroprocesorima bila odbijena od izdavača, Osborn se odlučuje da osnuje sopstvenu izdavačku kompaniju. „Neprihvatljiva knjiga“ je prodata u rekordnom roku od preko 300.000 primeraka. Godine 1979. Osborn je prodao svoju izdavačku firmu poznatom „McGrou Hillu“ (praksa američkih gigantata je da svaku uspešnu malu konkurentsku kompaniju na samom startu ili uništi ili otkupi). U periodu od 1979. do 1982. godine Osborn je radio u toku dana u svojoj novoj firmi „Osborn Komputjer“, dok bi u večernjim časovima, u skladu sa ugovorom o prenosu izdavačke firme, radio za „McGrou Hill“.

U svojoj novoj kompaniji na ključnim mestima zaposlio je čitavu grupu dvadesetogodišnjaka. Njegov potpredsednik odeljenja za marketing i prodaju je bila 25-godišnja Žoržeta Psaris. Ona je za svog prvog pomoćnika dovela 24-godišnjeg Lin Hagena, uz čiju pomoć je uskoro uvela popularni „osborn-ii“ portabl u svaku prodavnicu računara u Americi. U aprilu 1981. godine timu se pridružuje 24-godišnja Barbara Burdick kao stručnjak za odnose sa javnošću.

„Osborn-ii“ je postigao trenutni uspeh. Kako računar sam bez programa u rukama nestručnjaka ne vredi puno, svako ko bi kupio „osborn-ii“ dobijao bi i paket korisnih programa u vrednosti od 1.500 dolara. Samo dva meseca od početka isporuke, „Osborn Komputjer“ je ostvario svoj prvi milion. Na kraju prve fiskalne godine poslovanja kompanija je prodala računare u vrednosti od 100 miliona dolara, a broj zaposlenih je narastao na 800. U toku svojih putovanja po inostranstvu, Adama Osborna je rado pozivao u goste kre-



Junak kapitalističkog rada: Adam Osborn

diplomatskog, poslovnog i intelektualnog visokog društva. Osbornov entuzijazam je nosio sve. Svi su bili živo zainteresovani da investiraju u „budućnost“, u računarsku revoluciju, u „Osborn Komputjer“. Adam Osborn je bio predsednik kompanije sa stepenom ekspanzije kakav svet silicijumske doline do tada još nije video.

## Rastanak u szumaju

Prvi indikatori problema u „Osborn Komputjer Korporaciji“ su se pojavili u aprilu 1983. godine. U toku seminara koji je držao u Kolorado stigle su neugodne vesti. Izveštaji o rezultatu poslovanja u prošlom tromesečju bili su katastrofalni. Osborn nije mogao da poveruje. „Leteti“ dvadesetogodišnji menadžeri su ostali nemi, siomljena srca, bez posla i uništenih snova o milionima pre tridesete.

Uzrok brze Osbornove propasti, veruju mnogi, leži u neodgovarajućem rukovođenju. Adam Osborn lično nikada nije rukovodio sa više od 50 ljudi. Kompaniju sa skoro 1.000 zaposlenih bilo je potrebno voditi mnogo profesionalnije. U prvoj poplavi narudžbina, problemi rukovođenja su bili gotovo nevidljivi, sve dok im je postalo i suviše kasno. Prema Džonu Dvoraku, uticajnu kompanista „InfoWorld“... u pitanju je bilo rukovođenje stila „vasoniskog broda na raketni pogon. Kada se jednom raspao, njegovi motori su i dalje nastavili da zuje u pravcu Marsa.“ Problem je bio u tome što je „Osborn Komputjer“ započeo jednu revoluciju i u tom istorijskom pohodu zastao. Dok se rukovođač Osborna borilo da maksimalno profitira na „osbornu-ii“, kasnije, na mnogo boljej mašini istog tipa, „osbornu-ii“, novajilije na svoju portabl računara, kompanije „Kejpro“ i „Kompak“, fleksibilno su se nadovozale na „osborn“ novim idejama. Očigledna greška Osborna i mnogih proizvođača mikroročunara koji su u vreme potonuli sa Osbornom bila je u pogrešnoj proceni značaja odluke giganta IBM-a da izade na tržište mikroročunara.

Drugog avgusta 1983. godine, u restoranu firme, preostalih dvestotina zaposlenih je u szumaju održalo zadnji sastanak. „Os-

born Komputjer je bila kompanija ne samo sa ciljem da napravi novac — to je bio i san mnogih, ideal...“, priseća se danas Osborn. „Tog dana svi su saznali da je sa idealima gotovo, da je san završen.“

U vreme propasti Osborna, svima je bilo jasno da neko mora da propadne. Jednostavno, nije bilo mesta za stotine proizvođača računara. „Finansijski zemljotres“ je bio neminovao. Američki ekonomisti tvrde da je propast do zruče uspešnih kompanija dobar znak, znak zdravije ekonomije. Statistike su pokazale da u gradovima prosperitetna, kao što su bili Hjuston i Dalas, mnogo više firmi propada nego u gradovima stagnacije kao što su Klivlend ili Bafalo. David Bric, ekonomista sa MIT-a, tvrdi da zatvaranje firme pomaže ekonomiji da izvrši redistribuciju talenta i sredstava na njene najaktivnije i najproduktivnije punktove. „U našoj zivahnoj privredi, finansijske propasti su ekviperativno koji poučavaju druge kako ne treba i kako treba poslovati“, navodi David Bric.

## U mekom povezu

Nije bilo potrebno dugo vremena da se Adam Osborn ponovo oglasi u javnosti. Posle nekoliko meseci usamljenog sredinjava računa i balansa sa sumnjama u sopstvene sposobnosti, udružen sa Džonom Dvorakom, Osborn se prihvata pisanja knjige o avanturama „Osborn Komputjer Korporacije“. Pod zanimljivim naslovom, „Hipster Rast: Uspon i pad korporacije Osborn Komputjer“, knjiga je ponudila čitaocima zanimljivu priču i čitavu kolekciju top-traveža iz silicijumske doline. Ma koliko to bilo kontradiktorno, Osborn je svojom knjigom na temu kako je bankrotirao zaradio pravo malo bogatstvo! Bankrotiraj, dakle, da bi se obogatio.

Prema Džonu Hordejcu, profesoru ekonomije sa Dabson Koleđa, 65 procenata novih firmi propadne u toku prvih pet godina svoga postojanja. Od toga među propalima, 95 procenata se odlučuje na novi pokušaj. Adam Osborn, očigledno, spada u zadnju grupu. Nedavno je započeo sa svojom novom kompanijom. Ovoga puta glavni proizvod nisu jeftini i mali računari, već jeftin softver. Nova kompanija se zove „Paperback Software“ („Softver u mekom povezu“), zapošljava 15 uglavnom starih radnika, poznanika, a radne prostorije su dnevna i radna soba u kući Adama Osborna. Žoržeta Psaris je opet sa Osbornom, a tu su i ostali stari uvolci.

Niski udarac industrij mikroročunara sa dotada neviđeno jeftinom mašinom je, na neki način, došao gde „Osbornu“. Pitamo se da li će drugi niski udarac, ovoga puta industrij softvera, sa programima visokog kvaliteta po ceni od dvadesetak dolara, takođe doći glave Osbornu. Mora se priznati da ovoga puta ideja nije originalna. Poznato je da je Boriand Internacional bio prvi koji je započeo sa prodajom softvera profesionalnog kvaliteta po amaterskim cenama. Adam Osborn se nadovezuje ovoga puta i planira da programe prodaje kao knjige, po knjižarnama širom Amerike.

Sudeći po načinu na koji Adam Osborn operiše, do izlaska ovog broja „Računara“ („Softver u mekom povezu“) je verovatno već velika kompanija useljena u visoki oblatkoder u San Fransisku. A ako sve bude po klisheu, mač bankrotstva mu već visi nad glavom.





Dr Radomir  
A. Mihajlović

Razglednica  
iz Njujorka

## Matematika na računaru

Upotrebom novog softverskog paketa „MathCAD“ firme MathSoft iz Kembriđa, država Masačusets, moguće je opisati matematičke formule na PC-u u standardnoj matematičkoj notaciji kombinovanjem teksta, grafike i specijalnih matematičkih simbola.

„MathCAD“ je vrlo jednostavan program. Dovoljno je otkucati bilo gde na ekranu matematički izraz koji je potrebno izračunati, da bi „MathCAD“ automatski izračunao i prikazao na ekranu rezultat u vidu tabele sa brojevima ili — grafički. Promenom bilo kog parametra u nepoznatom izrazu vrši se automatsko ponovno izračunavanje. Uz bilo koji deo sa izračunavanjima moguće je dodati i tekstualni komentar.

Novi paket firme „MathCAD“ sposoban je da obradi probleme kako u realnom tako i u kompleksnom račun. Paket vrši automatsku konverziju jedinica i dimenzionalnu analizu. Između ostalog ugrađene su rutine za integraciju, diferencijaciju, brzu furijeovu transformaciju i interpolaciju kubnih segmentnih polinoma u niz diskretnih podataka (cubic splines). Paket je otvoren, pa je moguće ugraditi i sopstvene aplikacije.

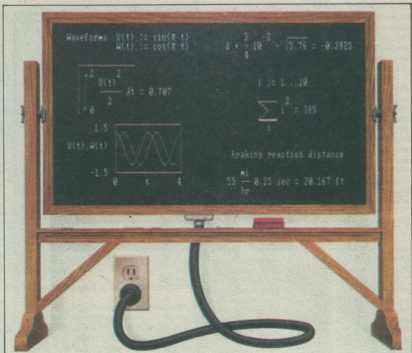
„MathCAD“ je moguće izvršavati na IBM-PC ili kompatibilnim računarima sa 512k RAM-a i grafičkom karticom. Ozbirno na mogućnosti, cena paketa je relativno skromna — iznosi svega 249 dolara.

## Bolji od Intela

Posle izvesnog perioda strpljivog čekanja na konačni ishod analize i procesa dokazivanja na tržištu, nekoliko značajnih proizvođača grafičkih kartica se konačno odlučilo da u svoje proizvode ugrade 32-bitni grafički mikroprocesor TMS34010 firme „Tekas Instruments“. Ovom odlukom je kao kandidat odabran Intelov grafički koprocesor 82786.

Mada je Tekasov mikroprocesor malo skuplji od Inteloveg, raznovrsnije mogućnosti i šira programabilnost su bile odlučujuće karakteristike. Primera radi, grafička kartica firme Conographics „ConoVision-2800“ nudi vrhunsku rezoluciju do 2880 sa 1024 tačkica (piksela).

Programabilnost čipa TMS34010 obezbeđuje izmenu karakteristika grafičke kartice



datodnim „zavijanjem“ hardverskih karakteristika u softver. Komuniciranje sa čipom se ne izvodi direktno već prema RAM-a u kome se nalazi softver, za izmenu identiteta“ grafičke kartice. Ovakvom primenom se grafički softver napisan za različite grafičke kartice i standarde može jednostavno prilagoditi na novu i moćnu karticu sa TMS34010.

Jedan od retkih kompanija koja se odlučila za Intelov čip, Quadram iz Džordžije, najavila je za februar ove godine svoju novu grafičku karticu označenu kao HPG. Nova kartica nudi rezoluciju ekvivalentnu rezoluciji IBM-ovog proizvoda PGC od 640 sa 480 tačkica i 256 od 16 miliona mogućih boja. Za razliku od PGC, kartica HPG će podržavati sve četiri vrste popularnih kolor monitora: CGA, EGA, Multi Sink i IBM Profesionalni Grafički Displej (PGD). Cena HPG kartice će, po svojoj prilici, biti približna ceni vrlo popularne EGA kartice.

## Digitalno memorisanje fotografija

Firme Canon, Nikon, Fujif, Minolta, Konika, Panasonic i Polaroid ponudile su, u različitim verzijama, tržištu specijalnu kameru za elektronsko generisanje fotografija. „Elektronski fotografski aparati“ su po spoljašnjem izgledu identični klasičnim foto-aparatima od 35 mm. Zbog ogromnog memorijskog kapaciteta potrebnog za čuvanje samo jedne fotografije u digitalnoj verziji, nove kamere memoriraju slike u analognom obliku na specijalnom minijaturnom video-rikorderu.

Glavni motor mikro-rikordera radi sa fantastičnih (i riskantnih) 3600 obrta u minuti. Za razliku od svakodnevnih video-rikordera, novi rikorder mirne slike ne radi sa trakom već sa magnetnim diskom. Nešto kao mikro analogna fleksibilna disk jedinica. „Jedan mikro disk prečnika 47 mm sa 51

koncentričnih tragova može da sačuva do 50 ramova slike, ili 25 ramova visoke rezolucije.

Prepuni mirko disk se preko ugrađenog modema može „preslikati“ na portabilni računarski masovnom memorijom. Sadržaj diska se može, takođe, preko telefona dostaviti, recimo, redakciji matičnog lista. Ovakvom primenom se vremenski razmak između snimanja značajnih događaja i objavljivanja vesti sa fotografijama drastično skraćuje.

Razgledanje elektronskih fotografija sa magnetnog diska je predviđeno preko specijalnih uređaja za reprodukciju koji se priključuje na kolor TV prijemnik. Reprodukcija signala slika je analogni, pa je zbog toga potrebno koristiti posebnu video karticu i softver za digitalizaciju video signala. Novi foto-aparat, sa brzinom blende do 1/1000 dela sekunde i elektronski sinhronizovanim flešom, proizvodi mnogo oštrije snimke brzo promenljivih scena od postojećih kamera i skaner uređaja.

## Standard za skaniranje grafike

U nameri da standardizuju tehnologiju za generisanje i reprodukciju slika pri elektronskom štampanju, softverske kompanije Aldus i Mikrosoft i hardverske kompanije DEST i Datacopy su se složile da podrže standard datoteka kod skaniranja i reprodukcije TIFF (Tag Image File Format).

Svrha TIFF standarda je da ohrabri korisnike da se zajednički pridržavaju istog načina organizacije, kodovanja i korišćenja digitalnih video informacija. U slučaju da se standard široko prihvati, mnoge softverske i hardverske firme bi radile na proizvodima koji bi podržavali datoteke direktno upotrebivši u kompjuterizovanom štampanju sa Mikrosoft softverom.



Datoteka uskladen sa TIFF-om je podejlena u tri osnovna dela. U prvom delu se nalaze informacije o važnim karakteristikama video informacije memorisane u trećem delu — delu za podatke. Deo za podatke je organizovan u polja čija je tabela nalazi u drugom delu datoteke.

Neka od polja sadrže podatke o strukturi podataka, rezoluciji, sadržaju dokumenta i fotometrički slike (vizuelno značenje podataka). Ovakvom organizacijom TIFF datoteke se minimizira količina potrebnih informacija dovoljnih za video podataka.

## Konačno video telefon

Novoosnovana kompanija Luma Telecom iz Santa Klare, Kalifornija, najavila je svoj jedinstveni proizvod — pravi video telefon sa cenom od 1.450 dolara.

Luma je vlasništvo moćnog japanskog koncerna „Mitsubishi“, koji je 1984. godine otkupio sve patente u vezi sa video telefonom od Atarija.

Osnovna komponenta video telefona je monohromatski ekran promera 7,5 cm na kome se prikazuje pokretna slika korisnika uređaja. Sliku generiše minijaturna TV kamera. Ako bi korisnik bio zadovoljan „svojim izgledom“, mogao bi da zamrzne sliku i pritiskom na dugme pošalje nešto ekvivalentno gore pomenutoj elektronskoj fotografiji. Prenos jedne slike kroz telefonsku liniju skromnog kapaciteta prenosa informacija traje između 1,5 i 5,5 sekundi — zavisi od kvaliteta željene reprodukcije i veličine slike. Po završetku prenosa mirne slike, na ekran se vraća pokretna slika korisnika.

Očigledno, usko grlo pri prenosu slike je skromna telefonska linija sa propusnim frekventnim opsegom od maksimalno 4.000Hz. Pokretna slika zahteva propusni opseg preko 1.000 puta veći. Prevedeno na jezik računara, potrebno je preneti oko 90.000.000 impulsa u sekundi. Skupim metodama kompresije signala, brzina prenosa se može smanjiti na 56.000.000 impulsa u sekundi. Da bi se smanjila cena, video telefon prenosi „relativno mirnu“ sliku, dok ugrađeni modem vrši minimalnu kompresiju signala pre prenosa.

Kamera ima fiksirani fokus na objekte udaljene 60 cm sa dozvoljenom varijacijom pozicije objekta od +/- 30 cm. Prikazana slika se može, po potrebi, odštampati na štampaču.

Očekuje se da će kupci video telefona biti prvenstveno poslovni ljudi koji imaju potrebu da pokazuju objekte ili dokumenta udaljenim sagovornicima. Popularna primena je tele-konferisanje počano umirenom slikom govornika. Slaba rezolucija slike je prepreka koja i pored relativno niske cene deluje obeshrabrujuće.

## IBM-PC XT/AT kao faksimil mašina

Gamma Link iz Palo Altoa, Kalifornija, nedavno je snizila cenu svog proizvoda za emulaciju faksimil mašine na IBM-ovim mikroracunarima, na svega 995 dolara. Proizvod je izašao u svojoj novoj, drugoj verziji pod nazivom GammaFax.

GammaFax osposobljava IBM-ov XT ili



AT računar da komunicira sa faksimil mašinama iz grupe-III. Osnovna komponenta je kartica sa modenom brzine 9.600 boda. Uz GammaFax karticu, neophodno je ugraditi u mikroracunar jednu od grafičkih kartica i obezbediti grafički štampač za reprodukciju. Jedna slika se memoriše kao jedna datoteka. Čisto tekstualna datoteka se tretira kao grafička informacija, a rekonstruše se metodom tačka po tačka.

GammaFax softver nije zaštićen i moguće ga je umnožavati. Po ceni od 1.200 dolara matična firma nudi softver za podršku Kanonovog skanera za automatizovano očitavanje i unos kucanog teksta u računar. Upotrebom skanera moguće je obavljati jednim uređajem dve funkcije: brzi unos kucanih podataka i prenos faksimil slika.

## Pol Alen osniva novu firmu

Posle oporavka od neke vrste tumora, Pol Alen, mormak koji je sa Bil Gejtson osnovao poznati „Microsoft“, krenuo je nedavno sa novim projektom. Nova firma se zove Asimetriks i prvenstveno se bavi izradom softvera za podršku postojećeg softvera iz „Microsofita“.

Osnovna namera Pola Alena je da razvija softver nove generacije. „Bazirani na dosadašnjim dostignućima u oblasti veštačke inteligencije i softverskog inženjeringa, naši programi bi trebalo da stave programera u mnogo prijatniju softver okolinu u kojoj je lakše rešavati probleme. Poređenje naših sistemskih programa i postojećih je slično poređenju asemblera i jezika C“, bez ulazjenja u više detalja, izjavio je u jednom od intervjua Pol Alen. Minimalna konfiguracija mikroracunara na kome će biti moguće izvršavati Asimetriksove sistemske programe je IBM-PC-AT. Alen je takođe pomeo nu da će originalni programi biti pogodniji za mašine bazirane na 80386 Intelovom supermikro-procesoru.

Prvi programi iz Asimetriksa će se pojaviti na tržištu u drugoj polovini 1987. godine.

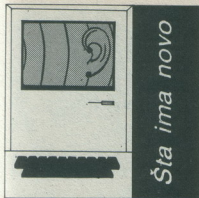
## Kolorisanje crno-belih filmova

Prošle godine je u SAD pojavila „ofarbanih“ klasika crno-bele kinematografije izazvala pravu buru različitih reagovanja. Većina filmskih kritičara je sa užasavanjem primila kič-novinu iz Holivuda. Za vlasnike tridesetast miliona video-rikordera koji svakog vikenda odlaze u „video šop na čoku“ i rentiraju po par kaseta kolorisani stari filmovi su pravo oševanje. O duševljenju vlasnika video radnji nije ni potrebno govoriti.

Zanimljiva činjenica je da se iza tehnologije kolorisanja starih filmova krije vrlo skromna elektronika — mikroracunar sa monohromatskim monitorom?

Firma Color Systems Technology iz Marine del Rey, država Kalifornija, koristi četiri „mekintosh“ mikroracunara kao stanice za kolorizaciju, dok petim „mekintoshem“ koordinira rad sistema. Obrada starih filmova se vrši uz pomoć para specijalnih video-rikordera koji su spregnuti sa mikroracunarima. Kompanija Colorization iz Ontarija, Kanada, koristi IBM-ov mikroracunar AT za rešavanje niza problema kolorizacije. U AT računar je ugrađena NTSC video kartica koja komunicira sa grafičkim procesorom Dubner CBG-II, koji se široko koristi u američkim TV stanicama za generisanje specijalnih video efekata.

Proces kolorizacije se sastoji od definisanju granica polja koje je potrebno ispuniti odgovarajućom bojom, sliku po sliku, sve dok data scena sa poljem traje. Automatizovano ispunjavanje polja monotono jednakom bojom daje otužan utisak. Međutim, sa druge strane, kolorisanje se uočljivo pojedinih objekata na sceni drastično povećava, tako da se ipak nešto dobija na kvalitetu filmova.



Šta ima novo

## HP — Opšti razvojni sistem

Hewlett-Packard već više godina unazad prodaje HP-64000 logički razvojni sistem čija je glavna namena razvoj softvera za razne mikroprocesore. Sistem je raspoloživ kao posebna radna stanica ili kao priključak na HP-serije 200, 300 i 500, a postoji i portabilna verzija. Sada se pojavila 32-bitna radna stanica 64130S sa 17-inčnim grafičkim terminalom, 80 Mby čvrstim diskom za backup-trakom i mogućnošću priključenja 5 vanjskih terminala. Stanica radi pod višekorisničkim HP-UX (proširena i poboljšana verzija operativnog sistema UNIX-V) i daje emulaciju i analizu sabirniča u realnom vremenu. Na stanice se priključuju posebni vanjski analizatori koji igraju ulogu integriranih osciloskopa, logičkih analizatora i merača performansi. Na raspolaganju su posebni pretprocesori koji se s jedne strane priključuju na vanjske analizatore, a sa druge na realni sistem za koji se softver razvija, tako da obezbeđuju pregled (kontrolu) analizu svake linije procesora. Pretprocesori sadrže assembler/disassembler za dati procesor, automatski analizator konfiguracije/performansi (nešto kao vanjski TRACE), simboličko mapiranje adresa, podataka i statusnih informacija te podršku interaktivnog rada sa radnom stanicom. Postojanje pretprocesora za neki mikroprocesor daje svojevrsnu sliku uspeha pojednog mikroprocesora, budući da HP spada među najznačajnije proizvođače merne, razvojne i računarske opreme. Tako se recimo pretprocesor za 80286 nikada nije pojavio dok je Motorolina serija 6800 i 68000 potpuno zastupljena kao i 8088, 8087, 8086 te Z-80, Z8001, Z8002. Najnovija je vest da je 80836 dobio svoj pretprocesor svega na pola godine od puštanja u komercijalni promet, što je najbolji pokazatelj velike podrške koju će uživati (ukoliko IBM ne požuri sa novom PC-mašinom može se desiti da on ovog puta pravi Compaq ili možda HP kompatibilce) među proizvođačima kako hardvera tako i softvera. (Ž. B.)

## Sve za „atari“

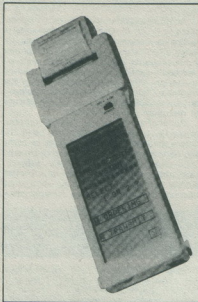
Pozната firma za proizvodnju periferijske opreme Cumana je proizvela seriju disk jedinica od 3,5 i 5,25 inča za „atari 520 ST“ ili 1040 ST. Postoje i kombinacije obe kategorije, kao i paketi od po dva ista drajva.

6/šta ima novo



## Amstrad u šarenišu

Grafička podrška za vaš „amstrad“ iz dana u dan jača. Najsvježija stvar na tom polju se zove MAX i omogućuje vašem „amstradu“ da nabaci pravi WIMP izgled. Tu su zgodne cake sa mišom, ikone, pomoć pri štampanju i drugim operacijama. Nove boje na ekranu. Sve to za samo 19.95 funti. Adresu ne znamo. (B. Đ.)



## Ručni terminal

Eponov ručni terminal sasvim definitivno predstavlja reakciju na „PO II“. On ima od 64 K do 256 K RAMa i 128 K ROMa, a sadrži LCD ekran osetljiv na dodir na koji stoji 12x14 karaktera. Kao kod siličnih napravica, tu je i RS 232 C port. (B. Đ.)

## Mali, malecki

Izgleda da su mali, malecki kompjuteri hit na tržištu. Drugim rečima, više firmi je najavilo svoje silične proizvode (da li je sve to prouzrokovao fenomenalni uspeh „Paion Organajzera II“?). Tako „Casio“ priprema malu ali moćnu mašinicu o kojoj ne znam preterano mnogo, osim da lepo izgleda, a obećanja su moćna. Pogledajte. (B. Đ.)

## Lepši partner za ples

Čini se da IBM ima dovoljno moći da drma industrijom čak i kad nešto ne uradi. Sudbina novog „Majkrosoftovog“ „milita-sking“ (višekorisničkog) operativnog sistema MS-DOS 4.0 je sasvim nezvesna zato što je IBM odbio da ga preuzme. Sad niko na tržištu ne želi da ima šta sa njim. Ne radi se o lošem kvalitetu. Jednostavno, IBM čeka na MS-DOS 5.0, koji je takođe višekorisnički orijentisan, ali za 80286, dok je MS-DOS 4.0 bio za 8086. Očekuje se da bi MS-DOS 5.0 mogao da se pojavi negde tokom leta ili jeseni 1987. (B. Đ.)

## Borlandov bejzik

Fascinantan uspeh programa „Turbo Pascal“ inspirisao je Borland da napravi svoj „Turbo BASIE“, koji na tržištu treba da se pojavi za nekoliko meseci. Borland najavljuje sasvim drugačiji pristup bejziku, kao i dodatne stvari u paketu. Bejzik će imati visokostruktuirani oblik nalik paskalu, a uz njega će se dobiti kompiler, ekranski editor i par drugih programčića za usputnu upotrebu. Sve to za samo 70 funti. Čekamo. (B. Đ.)

## Pravo uređivanje

Volite vredne stvari? Volite prave stvari? Nije vam važno koliko šta košta, bitno je da dobro radi svoj posao. Evo prave stvari za vas ako vas interesuje stono izdavaštvo. Legendarni Xerox, za koga se u našim krajevima (neosnovano) misli da je još samo jedna firma za proizvodnju kopir mašina, izbacio je na tržište stoni uređivač teksta ili ti „Xerox Documenter“. To je skup od nekoliko kopekih i lepih naprava. Tu je Xeroxova 8068 radna stanica, ogromni ekran u visokoj rezoluciji, tastatura, miš, hard disk od 10, 20 ili 40 MB, i laserski štampač koji može da radi kao foto-kopir mašina. Cena? Prava sitnica. Samo 7500 funti, a ako ste Jugosloven čak još 10% manje. Razmislite, pa skupite ti bednih 7500 funti. (B. Đ.)

# 24

# računari

specijalno izdanje časopisa „Galaksija“ izlazi jedanput mesečno izdaje BIGZ mart 1987. cena 400 din.

**Izdaje**  
Beogradski izdavačko-grafički zavod  
11000 Beograd  
Bulevar vojvode Mišića 17

**Telefoni**  
650-161 (redakcija)  
650-748  
650-528 (prodaja)  
651-793 (propaganda)

**Generelni direktor**  
Dobrosav Petrović

**V.d. direktor sektora Izdavačko-novinska delatnost**  
Antun Martić

**V.d. glavni i odgovorni urednik**  
Gavrilo Vučković

**Odgovorni urednik**  
Jova Regasek

**Likovno-grafičko uređenje**  
Mirko Popov, Vesna Raletić

**Redakcija časopisa „Galaksija“**  
Tanasije Gavranović, urednik  
Esad Jakupović, v.d. zamenik glavnog i odgovornog urednika  
Aleksandar Milinković, urednik  
Jova Regasek, odgovorni urednik  
Žarko Simović, sekretar redakcije  
Srdan Stojančević, novinar  
Gavrilo Vučković, v.d. glavni i odgovorni urednik

**Stručni saradnici**  
Radomir A. Mihajlović, Dejan Ristanović, Dušan Slavić, Nevenka Spalević, Anđelko Zgorelec

**Spoljna redakcija**  
Branko Đaković, Dejan Ristanović, Jelena Rupnik, Jovan Skuljan, prof. dr Dušan Slavić, Nevenka Spalević, Vlada Stojiljković, Zoran Životić

**Stalni saradnici**  
Nada Aleksić, Ninoslav Čabrić, Branko Đaković, Voja Gašić, Branko Hebrang, Željko Jurić, Radomir A. Mihajlović, Zvonimir Makovec, Blažimir Miše, Dejan Muhamedagić, Ivan Nador, Zoran Obradović, Miodrag Potkonjak, Dejan Ristanović, Jelena Rupnik, Dušan Slavić, Jovan Skuljan, Nevenka Spalević, Zvonimir Vistrička, Žarko Vukosa-vljević, Anđelko Zgorelec, Zoran Životić

**Izdavački savet „Galaksije“**  
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević, (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Veli-zar Maslač, Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofil-ović, Vidojko Veličković, Velimir Vesović, Milivoje Vuković

**Štampa**  
Beogradski izdavačko-grafički zavod  
11000 Beograd  
Bulevar vojvode Mišića 17

**Adresa redakcije**  
11000 Beograd  
Bulevar vojvode Mišića 17/III

**Rukopisi se ne vraćaju**  
**Pretpiata**  
Za zemlju: 4.400 din (jedna godina)  
na žiro-račun 60802-603-23264,  
Beogradski izdavačko-grafički zavod, 11000 Beograd

*Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika“ broj 26/72, ovo izdanje oslobođeno je poreza na promet.*

## sadržaj

2/razglednica izhujorka  
*Jedna američka priča*

6/šta ima novo

8/računari u izlogu  
*teška artiljerija*

10/najbolji u prošloj godini  
*miševi i mišice osamdeset šeste*

12/peek & poke show

14/programeri govore  
*„mrzim miševe!“*

16/organizacija računara  
*pc usluga na niskom nivou*

20/programski jezici  
*humani asembleri*

22/dejanove pitalice  
*kombinatorika u praksi*

27/umetak  
*sa bejzika na paskal*

48/tehnike programiranja  
*podešavanje tastature*

50/komercijalni softver  
*laserski genije*

54/računarski algoritmi  
*metodi gradljenata*

56/računarski algoritmi  
*osam krugova kredom*

60/osnove programiranje  
*brojni sistemi*

63/šta ima novo

64/razbarušeni sprajtovi



Razlikama između PC-a i AT-a smo se bavili u prošlim „Računarima“, u okviru prikaza nove Tendijeve linije PC kompatibilca — sećamo se da je IBM PC AT nastao kada je IBM zaključio da korisnici toliko proširuju stari PC da će ga vrlo rado zameniti novom mašinom koja je u startu opremljena svim tim drangulijama (i ponekim novitetom) i koja može da izvršava sve već kupljene ili napisane programe. Proizvođači PC kompatibilca su AT-a dočekali sa strepnjom: možda će njihove „male mašine“ postati zastarele i možda će IBM ponovo zadobiti čitavo tržište? PC kompatibilci su, međutim, i dalje interesantni, ali se pokazalo da tehnologija na kojoj je AT zasnovan (ma koliko, po rečima IBM-a, bila *Advanced* — napredna) nije imuna na kopiranje, pa su nastajali AT klonovi. Poput svih drugih kompatibilca, AT klonovi mogu da „napadnu“ IBM-ovu cenu i IBM-ove karakteristike; male firme su uglavnom napadale cenu, dok su se Olivetti, Wyse i ITT odučili da poboljšaju karakteristike i obore cenu. Ostaje da vidimo koliko su u tome uspeali.

## Olivetti M 28

Serijski M20 Olivettijevih PC kompatibilca je popularna i kod nas: poznajemo portabl kompjuter M21, „umerjeni“ M22, standardni M24 i prošireni M24SP. Olivetti je sredinom 1986. odlučio da dopuni liniju jednom AT kompatibilnom mašinom kojoj je dodelio broj 28 (slededi Olivettijev kompjuter će, slediće stepene dvojke, verovatno biti M36 ili M216). Prvi prospekti tvrde da će M28 biti višekorisnička mašina koja će raditi pod Xenixom ili nekim jačim operativnim sistemom, ali su donjine najave malo umerenije: operativni sistem je MS DOS, a korisnik je usamljen. Pokazalo se, naime, da mikroprocesor 80286, iako veoma moćan, nema arhitekturu koja bi mogla da zadovolji potrebe većeg broja korisnika tako da svaki od njih bude zaštićen od grešaka ostalih.

Pogled na našu prvu sliku navodi na pomisao da je M28 spolja identičan sa dobro poznatim M24; pogled na dimenzije, međutim, otkriva da je površina kutije nešto manja, a visina nešto veća — ove su razlike neophodne zbog AT kompatibilnih kartica. Otvaranje kutije otkriva štampanu ploču sa veoma malo komponenta. Poznavanje M24 će učiniti da ne budemo iznenađeni što vidimo računar bez mikroprocesora: kutija ima „duplo dno“ ispod koga je ogromna štampana ploča sa morem čipova koji, začudo, više „naglavačke“. Tu je, naravno, sesnaestobitni 80286, podnožje za aritmetički koprocessor 80287, megabajt RAM-a, 32 K ROM-a, dva prazna EPROM podnožja po 16 K, časovnik realnog vremena i mnogo drugih stvarčica koje uključuju i poveliku ULU. Serijski i paralelni interfejs su na štampali, što znači da ne zauzimaju ni jedan od 7 ekspanzionih slotova. Disk kontroler, ipak, mora da zauzme jedan slot, dok

se drugi koristi za priključenje hard diska što znači da nam preostaje 5 slotova: četiri 16-bitna i jedan 8-bitni.

Gratika je posebna (i posebno interesantna) tema. Osnovna ploča sadrži M28 *graphics controller* koji, prema želji korisnika, emulira IBM-ovu monohrom ili kolor (CGA) karticu. Uz računar može da se dokupi *enhanced graphics controller* (EGC), specijalna kartica koju je Olivetti dizajnirao u želji da ponudi rezoluciju 640\*400 u 16 boja koje se biraju iz palete od 72. Ista je kartica na raspolaganju vlasnicima M24 koji žele da se poluprofesionalno bave grafikom i koji smatraju da će umeti da preseku nekoliko vodova na štampi, zaleme nekoliko žica za pinove raznih integrisanih kola i preprogramiraju EPROM-e.

Nabavka kartice koja nudi visoku rezoluciju ne mora da znači da ćete tu rezoluciju zaista i koristiti pri radu sa komercijalnim programima: ako vaš omiljeni CAD paket nije opremljen odgovarajućim drajverom, izgled ekrana će, ukoliko brojite uopšte nazovemo izgledom, biti prilično smešan. Olivetti je, međutim, dobro poznata firma, što znači da će autori softverskih paketa imati dobar motiv da potroše neki sat rada na razvoj novih drajvera. Rezultati njihovog rada će, bez sumnje, biti na raspolaganju svima koji su nabavili originalne programe — ukoliko niste jedan od takvih (da li pirati čitaju „Računare“?), ulaganje 600 funti u EGC karticu može da predstavlja ulog koji će se isplatiti tek u daljoj budućnosti.

U cenu od 3400 funti je uračunato 640 kilobajta RAM-a, hard disk od 20 megabajta, jedna disk jedinica od 360 K ili 1,2 M (važje ograničenje koja smo opisivali u prošlim „Računarima“ — možda će diskete snimljene na M28 biti nečitljive na nekim „običnom“ PC-ju), monohrom monitor, serijski i paralelni interfejs. Tastatura je, kada se poredi sa raznim tajvancima, izuzetno kvalitetna, ali ipak zaostaje za originalnom AT-ovom.

Sve u svemu, Olivetti M28 je dobra, brza i savršeno kompatibilna mašina koja može da ima izvanrednu grafiku. Relativno visoka cena je rezultat posledica ogromnog broja čipova na štampanoj ploči — zapanjujuće je da mnogotruko moćnijim Compaq Deskpro 386 (prikazali smo ga pre dva meseca) ima delokalo manje komponenti!

## ITT XTRA XL

Dok se Olivetti pouzdao u dobru grafiku, ITT je ponudio klasičnog AT Kompatibilca sa moćnom periferijom: hard disk od 80 megabajta, dva flopija po 1,2 M, dva serijska i jedan paralelni port i sedam slobodnih slotova za dalja proširenja. Proširenja se ne ograničavaju na periferiju: RAM može da naraste na 1,64 megabajta ali se, jasno, i dalje koriste segmenti od po 64 K — drugačiju konfiguraciju mikroprocesor 80286 ne može da podrži.

Za razliku od većine drugih renomiranih firmi, ITT nije htio da se igra proizvodnje specijalnih grafičkih kartica: u bilo koji ekspanzion slot modela XTRA XL može da se priključi *bilo koja* PC ili AT komptibilna kartica, što znači da korisnik može da se zadovolji standardnom CGA ili monohrom karticom (i Herkules dolazi u obzir!) ili, ukoliko želi veći rezoluciju, da dokupi EGA Karticu i nov monitor. Moramo da kažemo da nam se ova feliksibilitet dopada, premda će, jasno, u nekim slučajevima biti relativno skupe.

Jaka strana modela XL je brzina: ne samo što mikroprocesor radi na kloku od 8 MHz, nego je i hard disk veoma kvalitetan pa je vreme pristupa podatku na spoljnoj memoriji oko 20 ms. Ponekad će se, međutim, dogoditi da vam brzina zastane: neki softverski paketi su zaštićeni rutinama koje su zasnovane na brzini kloka. Nema problema: uz XTRA XL dobijate MS DOS 3.1 koji je dopunjen komandom SPEED — otkucate samo jedan red i klik zakuca na 6 MHz.

Da bi se zapazile dobre strane modela XL treba pogledati njegove tehničke karakteristike. Sagledavanje loših strana je daleko jednostavnije: bacite pogled na grozan dizajn! Ne samo što je kutija toliko ružna da postaje smešna, već je njena visina poprilična, što znači da ćete, da biste ugledali monitor, morati da upravite pogled prema tavanici. ITT, doduše, isporučuje i specijalno postolje koje omogućava da CPU držite na podu ili ispod stola, što je dobro iz dva razloga: ne primećujete koliko je računar ružan i razglabavate se kada god umećete disketu ili resetujete kompjuter.

MS DOS, bar za sada, opslužuje samo jednog korisnika, što znači da za višekorisnički rad treba naći novi operativni sistem. Izbor, na žalost, nije velik: na raspolaganju je jedino Xenix. Oslono se smatra da Xenix može da opsluži najviše tri korisnika što nije sasvim tačno: ovo je ograničenje posledice arhitekture IBM PCA AT-a, koji ne može da podnese previše broj baterijskih ulaza-izlaza koje će zahtevati razni korisnici. Ukoliko se, sa druge strane, opremite I/O koprocessorskom karticom koja sadrži mikroprocesor 80186, XTRA XL će u teoriji lako opsluživati pet-šest terminala. Kažemo „u teoriji“, jer je nešto slično isprobano možda samo u ITT-ovim laboratorijama.

Sve u svemu, XTT XTRA XL je brz, moćan i veoma skup računar — 7000 funti je zaista previše čak i kada znamo da dobar deo ove cene otpada na hard disk od 80 M. Ukoliko ste spremni da izdvojite ovu sumu, savetujemo vam da je potrošite na 32-bitni kompjuter!

## Wyse PC286

Olivettijev glavni argument je grafika, ITT-ov brzina, a Wyse pokušava da ih prevaziđe po oba aspekta. Naredni redovi treba da odgovore na pitanje koliko je u tome uspeo, ali ne može da se sumnja da je

Već mesecima prve stranice „Računara“ pripadaju prikazu nekog PC kompatibilca pred kojim zastaje dah. Zapanjuju nas mogućnosti ali, na žalost, i cena. Vremena se, međutim, menjaju — pre samo dve-tri godine računar sa 32 kilobajta RAM-a i disk jedinicom od 100 K bio je san snova svakog hakera, dok su danas mnogima pristupačne i daleko jače mašine. Ako se ovakvi trendovi nastave (a nastaviće se), dani u kojima ćemo sebi priuštiti Olivetti M28, Wyse PC286 ili ITT XTRA XL nisu daleko!



## Tehničke karakteristike

Model	Olivetti M28	ITT XTRA XL	Wyse PC 286
Mikroprocesor	80286	80286	80286
Koprocessor	80287 (opc)	80287 (opc)	80287 (opc)
Clock (MHz)	8	6 ili 8	6 ili 10
RAM (K)	640	640	640 K
Maksimalan RAM (M)	1	1.64	1
Testatura	mehanička, 83 tastera;	mehanička, 84 tastera	mehanička, 84 tastera
Soft tastera	18	10	10
Ekran	Monohrom CGA ili EGC	Bilo koja IBM kartica	Bilo koja IBM ili WY-700
Znakovna rezolucija	80*25	80*25	80*25 ili 180*50
Grafička rezolucija	640*400, 16 boja	640*400	1280*800, 2 boje
Disk jedinica	jedna, DS DD	jedna, DS DD	jedna DS DD;
Kapacitet diskete	1.2 M	1.2 M	1.2 M
Hard disk	20 M	80 M	20 M
Interfejsi	RS 232, Centronika	2 * RS 232, Centronika	RS 232, Centronika
Real time časovnik	da	da	da
Slotova	4 16-bitna, 3 8-bitna	6 16-bitnih, 3 8-bitna	5 16-bitnih, 2 8-bitna
Operativni sistem	MS DOS 3.2, Xenix System V	MS DOS 3.1, Xenix System V	MS DOS 3.1
Cena (funtl)	3394 +582 (EGC).	7064 (sa CGA).	2250 +895 (WY 700).

PC286 po dizajnu višestruko superioran kako AT-u tako i klonovima koje prikazujemo: jednostavna ali elegantna kutija je visoka tačno koliko je potrebno da bi monitor bio u nivou vašeg pogleda, dok je umeštanje i vađenje disketa maksimalno pojednostavljeno.

Pogled na kontrolni panel modela PC286 otkriva par noviteta: ključ je dopunjen sijalicom koja svetli tek kada je on otkenut, druga LED dioda svetli dok računari radi, treća u toku pristupa hard disku, a poslednja indicira brzinu procesora: na raspolaganju je, naime, prekidač koji omo-

gućava izbor razne frekvencije od 10 ili 6 MHz.

Rasklapanje centralne jedinice otkriva poveću štampanu ploču sa mikroprocesorom 80286, podnožjem za 80287, 640 K RAM-a, časovnikom realnog vremena, šest šesnaestobitnih i dva osmibitna slotova za proširenja. Na osnovnom modelu su popunjena tri AT slota: jedan sadrži kontroler za floppyje, drugi kontroler za hard disk, a treći serijsko/paralelni interfejs. Grafička kartica će, kada je izaberete, zauzeti četvrti slot.

Izbor grafičke kartice zaista nije lak: Wyse PC286 će, naime, prihvatiti bilo koju PC ili AT kompatibilnu karticu, što znači da možete da koristite monohrom kartu, CGA

ili EGA adapter, pa čak i Herkules. Wyse je, međutim, dizajnirao i sopstveni video interfejs koji nudi profesionalnu grafiku 1280\*800 emulirajući pri tom IBM-ovu CGA ili monohrom grafiku. Kartica je nazvana WY700 i predstavlja proizvod vredan veike pažnje: nisu joj potrebni nikakvi posebni hardverski dodaci ili drajveri, kompatibilna je sa svim softverom za PC, a ipak nudi rezoluciju koja će, rekorno, zadovoljiti i profesionalce. Ukoliko, sa druge strane, proizvođač softvera uložio trud u pisanje WY700 drajvera, karakteristike računara PC286 će se dalje poboljšavati: WY700 može da prikaže osam punih ekrana iscrtanice u maksimainoj rezoluciji IBM-ove CGA kartice!

Od visoke rezolucije, jasno, nema mnogo koristi ako se ne opremimo dobrim monitorom. U osnovnu cenu računara PC286 je, zalista, uračunat u veoma kvalitetan monohrom monitor visoke rezolucije koji nudi stabilan i oštar tekst i preciznu grafiku; boja je, uz to, vrlo prijatna za oči, što je vrlo značajno ako nameravate da provodite sate i sate ispred računara.

Osnovni medij spoljne memorije je hard disk od 20 megabajta, a tu je i jedan floppy. U osnovnu kutiju može da se smesti još jedan floppy ili (po želji) još jedan hard disk, dok je priključenje eksternih diskova relativno komplikovano.

Wyse PC286 je, sve u svemu, lepo dizajniran i kvalitetan AT klon sa fascinantnom grafikom i, što je posebno interesantno, relativno pristupačnom cenom od 2250 funti.

## Kupiti AT ili sačekati?

Sva tri prikazana računara su bez ikakve sumnje veoma kvalitetna i vredna svake preporuke: čini nam se da od nosivosti i cene ide u prilog Wyse-ovoj novoj mašini koja se, uostalom, i poslednja pojavila na tržištu. Pitanje je, naravno, isplati li se u ovom trenutku investirati priličnu sumu u kupovinu AT kompatibilne mašine: sve je pristupnije mišljenje da je IBM PC AT zapravo slepo crevo PC standarda koje nema budućnost. Argumenata u prilog ovakvog mišljenja ima dosta: korisnici imaju sve manje razumevanja za segmente od 64 kilobajta i operativne sisteme zasnovane na „antičkom“ CP/M-u. IBM je, sa druge strane, oduvek gradilo svoj ugled na korišćenju provere tehnologije koja možda nije baš „na vrhu“ ali koja omogućava kupcima da računaju na dovoljne količine softvera i stručne literature. Čak i ukoliko se na tržištu uskoro pojavi novi PC zasnovan na 80386, AT neće biti mrtav: softvera za njega ima dovoljno za sve primene koje vam padaju na pamet!

Pojava nove IBM-ove mašine će, osim toga, verovatno izazvati drastičan pad cene AT klonova pošto će proizvođači svakako poželeti da „raščiste zalihu“. Treba se, dakle, strpeti par meseci. Ili možda imate par hiljada funti i čeznete da ih potrošite?

Dejan Ristanović

Najbolji  
u prošlog  
godini

# misteri i misice osamdeset šeste

**Kategorija:** Simulacije

**Pobednik:** TT Racer

**Firma:** Digital Integration



Verovatno bi o ovoj odluci moglo da se raspravlja, ali bi umesto toga bilo mnogo patetičnije reći da kompjuterši iz „Your Computers“ jednostavno vole da se trkaju. Uostalom, to je izvrsna igra kojoj svakako doprinosi i inovacija da je moguće povezati osam „spektruma“ i igrati se zajedno sa još nekim. Druženje na trkama.

**Kategorija:** Arkadna igra

**Pobednik:** Trailblazer

**Firma:** Gremiln Graphics



Mora se priznati da je ova igrice sa loptom koja skakuće po beskraju traci kroz svemir veoma zanimljiva. Nije joj odmglo ni to što se pojavila tek negde pred kraj godine. Pomalo se, ipak, atiče utisak da to nije samo nagrada za ovu super igricu nego i priznanje Gremilnu koji je u protekloj godini imao nekoliko zaista uspešnih igara.

**Kategorija:** Avantura

**Pobednik:** The Pawn

**Firma:** Rainbird



Članovi žirija u „Your Computeru“ su izjavili da je ovo bila kategorija u kojoj se bilo najlakše odlučiti. To verovatno i nije tako čudno kad se zna da ova igra ima najlepšu nepomičnu grafiku do sada videnu na „komodoru 64“. Priča je zanimljiva, igra je impresivna, ali je najvažnije u svemu tome fantastična grafika. Pogledajte i zadivite se.

**Kategorija:** Arkadna avantura

**Pobednik:** Antirlad

**Firma:** Palace Software



U objašnjenju za pobjedu ove igre žiri je napisao „Fantastična grafika, izvanredna muzika, dobra igra, privlačna grafika da vam je neoprijatno, a uz igru još dobijate i besplatan strip. Šta više da se poželi?“ Da, kažemo mi, šta više da se poželi? Nema tu svrhe za mnogo priče. Treba priznati da je Antirlad zaista najbolji u svojoj kategoriji.

**Kategorija:** Najinovativnija igra

**Pobednik:** Starglider

**Firma:** Rainbird



Ovo je ta igra o kojoj se toliko sluša u poslednje vreme. Svi su se nešto raspilavili oko nje. Dobro, igra je zaista lepa i tako to, ali to je još samo jedan mutacioni oblik dobre stare „Elite“. Inovacije? Ima ih, ali da li baš dovoljno za svu tu buku. Koje su to inovacije? Ah, da, prvi put pokretna 3D grafika na „spektrumu“.

**Kategorija:** Igre uopšte

**Pobednik:** World Games

**Firma:** Epyx/US Gold



Šta da se kaže? Odakle vam ovaj pobednik? Dobro, igra je zanimljiva, grafika je dobra i slično

neutrajne recenzije, ali zar nema boljih igara na tržištu? Ono što se svidelo žiriju je verovatno činjenica da je skupljeno osam egzotičnih igara iz belgo sveta. E, pa nama egzotične igre nisu toliko zanimljive, mi ih imamo i previše. Tačnije, nikakve druge i nemamo.

**Kategorija:** Grafički paket

**Pobednik:** Art Director

**Firma:** Mirrorsoft



Kada se prikazuju ovakvi programi, uobičajeno je da se kaže kako je program „izrazito jednostavan za korišćenje, ali zato moćan u primeni“. To je savršeno tačno i za ovaj program, ali to možda i nije pravi razlog što je baš ovo grafički paket godine. Postoje već kilograme sličnih za „meka“, o onima za „amigu“ se naveliko pisalo, čak ih ima i za PCja, a ovo je verovatno prvi ozbiljan takav paket za „atarija ST“. Naravno, odličan je.



**Kategorija:** Korisnički program

**Pobednik:** Fleet Street Editor

**Firma:** Mirrorsoft

Ovom programu je zgodno došla konverzija za PC klasu računara upravo u trenutku kada tajvanci prave ršum i kad je „amrad PC“ vruća roba. Jako dobar program za pravljenje novina (ako imate laserski printer rezultati su skoro profesionalni) uz pomoć pogodnog tržišta sa svim zaslužuje neko priznanje. Ovakvo veliko priznanje je verovatno pre rezultat siromaštva u ovoj kategoriji nego nekog astronomskog kvaliteta upravo ovog programa.



Kao i prošlih godina „časopis „Your Computer“ je objavio svoju listu najuspešnijih za 1986. godinu. Sasvim normalno, kada se uzme u obzir orijentacija „Your Computera“ najviše prostora je posvećeno igrama, ali ima i sasvim drugačijih kategorija. Po čemu je to „Your Computer“ zapamtio godinu koja je upravo prošla? Umesto uobičajene „Razglednice iz Londona“ (Anđelko Zgorelec je u poslu do guše jer pokreće dva nova časopisa), pripremili smo skraćeni pregled kroz prizmu Branka Đakovića.

Kategorija: Kompjuter godine

Pobednik: Amstrad PC1512



Da li bi ovo trebalo da iznenadi bilo koga? Amstrad PC je dojahao na talasu promena na PC tržištu i dobro ih iskoristio. Da li ima boljih mašina? Puno. Da li ima puno nejasnih stvari vezanih za kvalitet „amstrad PC“? Sasvim sigurno. Da li je ovo dobar izbor za mašinu godine? Svakako. Ovo je kompjuter o kome se pričalo, koji je bio u centru pažnje, koji je ustalao i duhove i tržište. Da li vam se sviđa? Da li vam se ne sviđa? Sigurno vas ne ostavljaju ravnodušnim. To je ono.

Kategorija: Napravica godine

Pobednik: Psion Organiser II



Stvarčica za koju smo mi navijali. Pravi mali džepni kompjuter za koji ljudi iz „Your Computera“ kažu da je postao zaštitni znak poslovnih ljudi koji voze BMW-e. Oni ne znaju da su ovdašnji vozači BMW-a uglavnom vlasnici hamburgerdžinica (jel se tako kaže?) i da nikad nisu ni čuli za „Psion Organiser II“. Uglavnom, ako ste pažljivo čitali „Računare“, primetili ste da nam se sviđeo Organiser II i da preporučujemo da ga isprobate. Sad nas podržavaju i kolege iz „Your Computera“. Hvala kolege, znali smo da smo u pravu.



Kategorija: Periferijska naprava godine

Pobednik: Thingi



Ovo je da ne poverujesi! U redu, ta uvrnuta plastična pločica za držanje listova sa kojih čitate je zaista zgodan štos. Mnogo zgodan štos. Ali proglasiti to (i još jednom: TO !! ?) periferiskom napravom godine je čista perverzija. Sad mi je tek žao što se nisam kandidovao za titulu programera godine. U ljubavi, ratu i računarstvu sve je dozvoljeno, ali ipak treba naglasiti da se u ovoj kategoriji radilo o intenzivnom zamoru međusobnih mišića celog žirija.

Kategorija: Džojstik

Pobednik: Speed King

Sad je lako reći da je ovo dobar džojstik. Ovo jeste dobar džojstik, ipak, nema mnogo svrhe to govoriti. To će da primeti par snobova koji moraju da imaju najbolje iz svake oblasti, a većina domaćih privrženika će i dalje da puca i puca iz svojih većitih Quck Shotova II. Ovaj džojstik (misli se na Speed King) može da izvršno posluži kod CAD-a. Što je CAD, pitaju domaći ljubitelji malih zelenih. Nema veze. Ovo je dobar džojstik.

Kategorija: Ličnost godine

Pobednik: Alan Sugar



Da, Alan Sugar je ličnost godine. To i nije neko iznenađenje osim ako živite negde na Polineziji. To je onaj strašni biznismen koji je napravio strašan posao od Amstrada, pa zatim kupio Sinkler, pa refrizirao „spektrum“, pa onda drmnuo šakom o sto i izbacio na tržište „amstrad PC. Sve je to on uradio. Strašan čovek, a? Samo, da li činjenica da je on jedan fantastični trgovac koji se sasvim slučajno bavi računarima znači da je opravdano što je on ličnost godine u jednom računarskom časopisu? Da. Računari se možda još uvek mere bajtovima i sličnim trivijalnim stvarima, a računarska industrija kao meru uzima pare. Da, ovo je jedan gigant računarstva. Rame uz rame za Alanom Turingom. Alan do Alana, svi bez mana. Malo mi se vrti u glavi.

Kategorija: Softverska kuća

Pobednik: Hewson

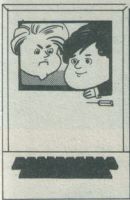


Jako je šarmantno što drage kolege iz „Your Computera“ nisu trošile svoje dragoceno vreme da razgovaraju o firmama koje ne proizvode igre. Šta ima Hewson (nekada Hewson Consultants) iz sebe u ovoj godini? Najbolje stvari su Unidrum (sviđa mi se), Quazatron (baš lepa stvar), Pyracuse (bjah), Iridis Alpha (ništa naročito) i Firelord (zgodno). I to je ono zbog čega je ovo softverska firma godine? Ako sam dobro shvatio, od ove godine ćemo u našim tečaj procesorima imati opciju pucanja na vanzemaljske, a ako je to neki jači tečaj procesor možda i hamba? Čestitmo sa uzdržanom usom.

Branko Đaković

YEAR'S BEST  
1986

11/najbolji u prošloj godini



Peek & poke  
show

Kutak za lude i  
zbunjene

Nesrećni Cep  
Cep i njegova  
zlodjela

Nastavljam sa našim malim serijalom koji prikazuje sve one istorijske trenutke u kojima je nama (meni naročito) veoma draga veštačka inteligencija uticala na život ljudi i tako pomogla da se budućnost primakne za još par dana, a haos produći.

Danas ćemo našu pažnju posvetiti onome što je dugo bilo skriveno pod velom ljudske zavisti — činjenici da su kompjuteri kreativni. Veštačka inteligencija je oduvek bila stavljana pod znak pitanja kod te jedne tačke — da li će kompjuteri umeti da stvaraju. Ukratko: moguć! Ja to najbolje znam, ali bilo je teško dok se to prvi put nije shvatilo.

Prvi korak u istoriji kompjuterske umetnosti je predstavljao stvaranje VI kompjutera poznatog pod nazivom Brand KJ Scriboman. To remek-delo automatskog pisanja (u pravom smislu te reči) je uspeo da za nešto manje od godinu dana napornog rada stvori sedam romana, tri zbirke pesama, osam drama, dve knjige eseja, jedno nedefinirano pornografsko delo, više stotina novinarskih članaka i dva računarska novinarska članka. Dokaz kvaliteta su bile i dve NIN-ove nagrade, pisma oduševljenih čitalaca i poziv opštinske poreske službe na omanji razgovor.

Bilo je logično za očekivati da će taj divni izum naići na široku podršku i da će čemo uskoro imati puno, puno umetnosti, ili bar veštački inteligentne umetnosti. Ali konstruktori Brand KJ Scribomana se nisu mučili mnogo da ga usavršavaju. Umesto toga, napravili su prvog pravog kompjuterizovanog književnog kritičara koji se pojavio pod šifrovanim imenom Cep Cep.

Pre pojave Cep Cepa većina ljudskih kritičara se odnosila prilično mekano prema delu Brand KJ Scribomana. Jednostav-

## Draga Saveta

### Najbolja programerka

Drugarice Šljukić, potrebna mi je vaša pomoć! Stavljena sam u tešku situaciju. Moja ćerka Dragica ide u srednju usamerenu školu koja neguje neko kompjutersko usmerenje. To je sve jako fino i ja nisam imala ništa protiv toga do sada. Sad su nastupili problemi. Moja ćerka je najbolja programer u razredu. To je dovoljno do toga da joj zavide sve devojčice, a da je svi muškarci definitivno mizis. Možete li zamisliti to? Najbolji programer u razredu je žensko. Odmah da vam kažem: Ne želim da moje dete postane kompjuterska sedesilica. Želim da živi normalan život, da ima normalna iskustva, pa makar to bile i ove face i slično, ali ni jedno muško dete neće da pogleda moju Dragicu zato što je tako dobra sa tim prokletim kompjuterom.

Pokušala sam da joj objasnim čega se bojim, ali me ona ne sluša. Možda još nije kasno, možda će na studijama sa nekim kolegom učiniti nešto, ali šta ako upiše studije gde će opat biti najbolji programer i gde će je opat zamrzeti? Pomozite mi, recite mi kako da ćerci pomognem da joj ove stvari ne smetaju u onim stvarima. Još jednom vas molim, pomozite.

Marta Mirković, Beograd

Draga Marta, odmah da vam kažem da nema razloga za zabrinutost. Anketi i usmereni psihološki testovi su pokazali da se stvari o kojima ste pričali unekoliko manju kasnije, tokom školovanja. Košeg vaše ćerke će promeniti svoj stav prema njoj negde pred kraj studija, kad shvate da će oni, a ne ona, na kako joj bile dobre ocene, biti rukovodnici računarskih centara i glavni programeri. To je otprilike faza poznata pod nazivom zakasneli računarski pubertet.

Saveta Šljukić, dr pr psi



no nisu bili u formi. Kako kritikovati nekog ko nije čovek? Jako teško. Tako je delo poznatog nam B KJ S-a ostalo bez kritičke analize i sličnih degradacija. To jest, ostalo je do pojave Cep Cepa.

Cep Cep nije patio od sitnih ljudskih slabosti. Uspeo je da pomoću ubitačne logike, divno sintetizovanih argumenata i obilja izvorne kompjuterske mržnje sasvim detaljno obradi delo Scribija Šekspira. Bila je to tako uspešna kritička analiza da je krv (kompjuterski govoreći) letela na sve strane.

To nije moglo proći bez posledica. Ista grupa konstruktora koja je bila zadužena i za dosadašnju književnu gungulu je tada, da unese malo pravde u svet bitne književnosti, konstruisala novog kritičkog veštački inteligentnog stvara koji je dobio ime Hartart 86. Hartart 86 je imao stav dijametralno suprotan stavu već glasnog Cep Cepa. Njegov hvalospevi u čast fantastičnih Brandovih dela su bili tako sladunjav i neukusni da čak ni ljudska bića nisu mogla da ponove tako nešto.

Sve to se malo iskomplikovalo kada je Brand u nastupu odbrambene kreativnosti (na kraju krajeva to mu je bila funkcija) napisao odgovore kritikama Cep Cepa i Hartarta 86. Odgovor Cepu je bio surov i umereno duhovit, a kratka nota Hartartu je bila ponižavajuća i podsmehljiva. Cep Cep je napisao još jedan napad koji je bio još otravniji od prošlog u kome je pozivao na

zaštitu Naših Pravih Kompjuterskih Vrednosti od Onih Koji Skreću Sa Našeg Kompjuterskog Književnog Puta. Hartart 86, se sa svoje strane, još više povukao u sebe i napisao jedan tužan traktat o naglom procvatu i propasti Moderne Kompjuterske Umetnosti.

Trebalo bi naglasiti da je čitava ova tuča obilato zabavljala kulturne poslenike celog sveta, ali samo do trenutka kad Brand KJ CEP Cep i Hartart 86 nisu u svojoj žustroj diskusiji počeli da potkačinju i ljudske pisce i kritičare. Tada je počeo rat. Laboratorija za veštačko pisanje i brisanje je pod hitno izbacila na tržište novi elektronski mozak koji je štito ljudske pisce od kompjutera, a ubrzo zatim i kompjuter koji je štito kompjuterske pisce od kompjutera koji je štito ljudske pisce. Ubrzo se javila potreba za kompjuterom koji štiti svakog od svakog i, odmah zatim, kompjuterom koji sve napada.

Prilično je nepotrebno reći da se čitava situacija pomalo iskomplikovala. Čudnim sticajem okolnosti svi uučveni u raspravu, a to je bila većina kompjuterskih i ljudskih pisaca, nisu pisali baš previše, što je za neke bilo značajno poboljšanje u kvalitetu.

Tada je na scenu stupila Svetska Pisačka Organizacija koja je po kratkom postupku stavila van zakona kompjutersko pisanje i tako nas sve zajedno vratila u dobra stara opuštena vremena nesistematičnog pisanja.

To je kratka predistorija, dragi čitače, zašto danas svi mikrokompjuteri jednostavno ne mogu da pišu, pa čak ni da služe za ozbiljno pisanje. Svi oni koji su na njima pokušali da kreativno rade bili su žrtve deformacije logike rečenice i smisla teksta i lepote izraza. To je posledica preprogramiranja kojem su podvrgnuti mikroprocesori svih malih kompjutera da nikad ne narastu do opasnih kibernetičkih pisaca kakvi su nekada bili. Zato oporastajte onima koji ružno pišu na svom kompjuteru — njihov mikrač je protiv njih, ali doći će bolja vremena i mi ćemo pisati zajedno.

Istoriske kompjuterske ane prelistavao, iz njih preplivao i sve to napisao P.A. Marvin

# Čip Pobody Agency

## Baruštine recesije

Sledećih nedelju dana bićemo domaćini filmskoj ekipi koja pravi poznatu prirodno-obrazovnu seriju „Opstanak“. Kako saznajemo, njihov jedini zadatak u ovom podneblju je da se snime uverljivo stariju (i još jednoj ugroženoj vrsti); jugoslovenskim programerima. Serija se snima bez zaštitnih kaveza i sa puno deviznih mamaca.

## Slatka budućnost

Na savetovanju održanom u kompjuterskom centru Dubrovnika Savezni sekretar za unapređenje računarstva i sočlenje ulica je pri kraju svog dugog i savršeno čistog teksta izjavio da se „pred nama tek nalazi slatka kompjuterska budućnost“.

Smisao te rečenice je za mnoge (bar za one budne) bio misterija sve dok se nije saznalo da će firma „Bajtni slatkiš“ iz Donjeg Mrđajevca početi, po ideji preuzetoj iz prošlog broja „Računara“, da proizvodi čokoladne slatkiše u obliku kompjuterskih čipova. Njam njam!

## Godina lagodnog življenja

Saznajemo da će 1987. godine u Jugoslaviju biti proglašena za Godinu Kompjutra i da će biti organizovana prava gomila prigodnih manifestacija. Kao moto za čitavu godinu i sve te manifestacije uzeta je rečenica iz legendarnog romana „Foliranje do besvesti“ Brand KJ Scribomara: „Nema više potrebe da kaskamo za svetom. Mi smo već sa njim, on nas nosi u zubima.“

## „Politika“ kao računarska sudbina

Još jedno ekskluzivno otkriće. U neformalnom razgovoru sa našim specijalnim i tajnim dopisnikom najpoznatiji američki računarski komentator Čipendey Blabing je izjavio da se ne usuđuje da donese nikakav zaključak pre nego što detaljno izanalizira najsvježiji primerek rubrike Malih oglasa beogradske „Politike“. On tvrdi da je mnoge značajne stvari u prošloj računarskoj burnoj godini pogodio unapred pažljivim posmatranjem računarskih oglasa u istoj rubrici i njihovom semantičkom analizom.

Ovaj neujumljeni broj „Peek & poke show-a“ pripremljiv sa Leonard C. Pecov Mika, Vuč Karadžić, Saveta V. Šljukić, PAM i Branko M. Đaković. Niko od nas ne odgovara za sračne smetnje i slične reakcije nastale usled pogrešnog korišćenja teksta.

## Slobodni baferi

## Uči me nežno

„Kad se vidi kakve stvari ljudi čine, nije ni čudo što svi iščekuju veštačku inteligenciju.“ — Bufalo Bil.

Odavno se ništa nije događalo sa našim školstvom. Već preko pola godine nema nikakvih promena u njemu i svima je već počelo da biva dosadno. Evo, naiznad, promene i to u vezi sa računarstvom. To vredi pogledati.

Od jeseni će sve škole na području Srbije uz pomoć svetog trojstva Zakona, Plana i Programa među svoje ostale predmete tiho umuvati i računarstvo. Ono će biti mešano u časove OTP-a (III?), i to uz obavezan minimalni broj računara i prateće opreme. Počeće nastava, kinci će učiti, znaće više i biće spremniji za ostale škole koje ih čekaju i tako dalje.

Pričica o tome kako svi mi želimo da naša deca uče o kompjuterima i da tako budu sposobniji, za savremeni svet je stara taman dovoljno da se izliže od upotrebe. Znači, mi želimo da deca uče o kompjuterima. Samo, nekako to nismo ovako zamišljali.

Promena nije dobra ako nije štura, brza i haotična. Tako glasi dobro uveštani zakon naših bezbrojnih impotentnih pokušaja da bar malo poboljšamo naše krpljeno i razbučeno školstvo. To pravilo je, čini se, jako dosledno primenjeno na ovu, računarsku promenu.

Da li se zna ko će deci predavati računarstvo? Kad bi SIŽovi obrazovanja imali pare koje nemaju (setite se da je u ovoj godini zaočinstvo plata prosvetnih radnika već u planu 10% iz privrede), možda bi mogli da se zaposle neki od onih silnih nezaposlenih mladih ljudi. Ovakvo, računarstvo će predavati nastavnici matematike, tehničkog i sličnih predmeta. Ljudi koji su na brzinu obučeni, tek će biti obučeni ili nikada neće biti obučeni. Jesu li oni krivi što im se to gura u ruke? Naravno da nisu. A učenići?

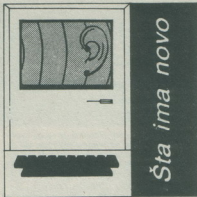
Da li se zna koji će kompjuter biti u školama? Ne zna se, ali će se veoma brzo odlučiti. Ako se prelista leksikon jugoslovenskih i badavadžijskih izraza, saznaje se da „veoma brzo“ označava period vremena „između deset minuta i tri miliona godina“.

Da li postoji neki udžbenik ili (možda čak i) udžbeničić? Ne baš. U pripremi su. Bilo bi ugodno posediti da se „priprema“ udžbenika za usmereno obrazovanje protegla na citave četiri godine.

Da li su škole i nastavnici obavešteni? Da, ako čitaju iste novine koje i ja. Tačnije, neki su znali i ranije, ali su smatrali da se radi o neuspehom crnom vicu. Sad će morati da promene mišljenje, ili bar smisao za humor.

Još jedno pitanje, na samom kraju. Da li se zna ko je zamislio ovaj lepi kolačić? Ne zna se, uglavnom zato što se radi o skromnim ljudima koji u potpunosti pripadaju ovom dobu kolektivne odgovornosti, nulte kreativnosti i konačno postignutih uslova za nastanak veštačke inteligencije.

Neka nam CPU pomogne.



Šta ima novo

## Miran mlaz

U RA18/19 čitali ste prikaz „Mlaza Koji Misi!“, to jest HP-2225 ThinkJet printera, a sada se pojavio njegov sledbenik „Miran Mlaz“ HP-2227 QuietJet, koji ima ugrađena dva međusloja (RS232C i Centronics), dvostruko veći bafer, brzinu od 192 karaktera u sekundi (draft) i 48 karaktera u sekundi (NLQ). U NLQ modu postoji potpun pristup masnim i podvučenim slovima te gornjim i donjim indeksima (superscript i subscript — ovo je posebno značajno za izvođenje nekih finesa i kondenzovanje ispisa kada je pravu PLRQ nužno potreban, a na većini ostalih printera nedostupan).

Za razliku od svog prethodnika „Miran Mlaz“, prihvata sve formate papira uključujući sa A3, iako je vrlo malih dimenzija (572 mm x 214 mm x 118 mm). Sledeće poboljšanje se ogleda u dodavanju trećeg grafičkog moda tako da pored standardnih 96 x 96 i 192 x 96 ima i 192 x 192 tačka po inču (jedna tačka je približno 0,13 mm) što daje još veće grafičke sposobnosti (pogledajte slike u RA18/19 da se posredite kakve su bile do sada). Unutar kućišta je ostavljen prostor za ROM i RAM čipove, što korisniku omogućava dodavanje „custom“ skupa karaktera bez izmene glavnog ROM-a ili takozvani „download“, tj. punjenje posebnog RAM-a definicijama karaktera pri uključivanju štampača (znajući proizvodnu filozofiju ovog proizvođača mogli bismo očekivati i pojavu takozvanog kontinualnog RAM-a, koji se neće brisati bar jedno dve godine po isključivanju). Ovaj je način u poslednje vreme postao popularan budući da omogućava brzu, jeftinju i vrlo fleksibilnu izmenu odnosno proširivanje osnovnog skupa karaktera (kojih ovde ima 132).

Šum koji stvara printer je svega 48,5 dB (broj vrata bi za ovo bio -1) i uglavnom potiče od mehanizma vođenja glave i papira (ugrađen je i traktorski i frikcioni sistem). Glava-reservoar je ista kao i ranije, s tim što sada imamo na raspolaganju četiri boje: crnu, crvenu, plavu i zelenu. Glave možemo menjati usred štampanja, pa ovaj sistem obećava možda skoriju pojavu „ColorJet“ printera. Printer je težak samo 4,27 kg a troši maksimalno 18 vati. Hewlett-Packard nastavlja, dakle, seriju izvanrednih i jevnitih štampača na radost svih onih koji su željni kvalitetnog otiska, maksimalne fleksibilnosti i tišine u radnoj sobi. (Ž. B.)



# Programeri govore

## Džef Raskin

### konstruktor računara „mekintoš“

Džef Raskin (Jef Raskin) je u pravom smislu reč čovek sa hiljadu talenata, ali će njegova dostignuća i uspehi verovatno uvek biti zasenjeni činjenicom da je on osmislio i sproveo u delu projekt „mekintoš“. Bivši dirigent Kamernog orkestra opere u San Franciscu; drži patente u dizajnu ambalaze u oblikovanju aviona i u elektronic; umetnik čija su dela izlagana u njujorškom muzeju modernih umetnosti i Okružnom muzeju u Los Anđelesu i, konačno, rukovodilac firme Information Appliance... **Roden je u Njujorku 1943. a on sam naglašava da je tada rođen i prvi digitalni kompjuter.**

Bio je jedan od pogonskih kreativnih ljudi u Eplu, iako je zvanično prvo bio zadužen samo za štampu i priručnike, a tek potom i za projekte u razvoju.

Aktivno je studirao filozofiju, umetnost i kompjuterske nauke. Predavao je kompjuterske nauke i na univerzitetu, a bio je i direktor kompjuterskog centra. Bogata karijera i puno iskustva.

— Poznati ste kao tvorac Eplovog (Apple) „mekintoša“ (Macintosh). Koja je vaša uloga u tome?

Te godine, 1979, Epl je radio na „lizi“. Verovili ili ne, „liza“ je prvobitno zamišljena kao generator karaktera. Ja sam bio rukovodilac razvojnih projekata u Eplu i bio sam prilično nezadovoljan „litzom“. Bila je veoma skupa i zato nisam smatrao da je dobro što se Epl tekmiči sa DEC-om, Data Generalom i IBM-om na polju minikompjuterskih cena.

Dok sam bio gostujući predavač u Stanfordskoj laboratoriji za veštačku inteligenciju, razvio sam dosta vremena u Kserokovom razvojnom centru u Palo Altu. Mislio sam da je ono što Kserokov PRAC radi sa bit mapiranim skrinovima, generalizovanom tastaturom i grafikom izvanredno. Zato sam se potrudio da promienim neka mišljenja u Eplu i „liza“ je pretvorena u bit-mapiranu mašinu. Pomogao sam da se spoje Kserokov i Epl. Negde u to vreme, Kseroka je izdao oko 10% Eplovih akcija.

Ono što sam ja predložio je bio kompjuter koji će se lako koristiti, koji će moći da meša tekst i grafiku i koji će se proizvoditi za oko 1000 dolara. Stiv Džobs (Steve Jobs) je rekao da je to luda ideja, da se to neće proizvoditi i da Epl neće da ima ništa sa tim. Pokušao je da obori taj projekt.

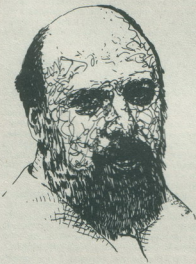
Zato sam zaobišao Džobsa i posetio tadašnjeg rukovodioca Majka Markula (Mike Markula) i popričao sa njim o mojoj ideji. Na sreću, i Markula i tadašnji predsednik Epla Majk Skot (Mike Scott) su naložili Džobsu da me ostavi na miru.

Unajmio sam ljude za projekt: Bada Tribla (Bud Tribble), Brajana Havaarda (Brian Howard) i Barela Smita (Burrell Smith). Otišli smo u savsno novo prostorište izgradili prototip „meki“ i njegovog softvera i uspeli da proradi.

Kasnije, pošto je pruzao stvari, Džobs je izmislio priču o tome kako je projekt „mek“ bio „piratska operacija“. Nismo pokušavali da držimo projekt dalje od Epla, kako je on rekao; imali smo veoma dobre veze sa ostatak Epla. Pokušavali smo da držimo projekt dalje od Džobso-

vog mešanja. Prve dve godine Džobs je pokušavao da uništi taj projekt zato što pojma nije imao o čemu se tu radi.

Originalni „mekintoš“ je bio pažljivo i racionalno projektovan. Konačno je svako u Eplu shvatio da je to Eplova glavna nada za proizvod koji je trebalo da dođe posle „Epla II“. Tada je Džobs preuzeo sve. Jednostavno je došao i rekao „Ja preuzimam „mekintošev“ hardver; vi možete da nastavite sa softverom i priručnicima“. Izbio je naš originalni softverski dizajn i učinio „mekintoš“ kompatibilnim sa „litzom“, a takođe je imitirao i da se koristi miš. Mašina je postala veća, komplikovanija i mnogo skuplja. Onda je, posle nekoliko meseci, Džobs rekao: „Preuzimam softver; vama ostaju priručnici.“ Ja sam mu rekao: „Možeš da uzmeš i priručnike.“ — otišao! To je bilo u maju 1982. On i Markula su mi rekli: „Molimo te, nemoj da ideš. Daj nam još mesec dana i pripremišmo ti ponudu koju nećeš moći da odbiješ.“ Tako sam dao Eplu još mesec dana; oni su meni dali svoju ponudu, a ja sam je odbio.



„Nisam želeo kompjuter. Želeo sam nešto što bi mi radilo posao“

— Čini se da ste prilično osetljivi kada su u pitanju priznanja za svoj rad?

Nije mi namera da preuzmem sve zasluge za projekt „Mekintoš“. To je bio timski rad. Kada bi Džobs tražio priznanje samo za ono što je zalata i uradio, to bi bilo sasvim dovoljno. Ali, on pokušava da preuzme i zasluge drugih, a to nije baš najbolje.

Bilo mi je prilično zabavno nedavno objavljeni članek u časopisu „Newsweek“ u kome je on izjavio: „Imam ja još dobrih projekata pred sobom.“ Nije on nikada imao svoje projekte! On nije projektovao ni jedan jedini proizvod! Stiven Vozniak (Steve Wozniak) je projektovao „Epl II“. Ken Rotmiler (Ken Rothmiller) i drugi su projektovali „litzu“. Moji tim i ja smo projektovali „mekintoš“. Vendel Sanders (Wedell Sanders)

je projektovao „epi III“. Šta je Džobs projektovao? Ništa.

— Kada ste napustili Epl, da li ste mislili da ćete ponovo raditi u tom poslu?

Mislio sam da neću raditi ništa što je tako glupo kao što je rad u Silicijumskoj dolini. Bio sam umoran od rada sedam dana nedeljno.

Posle sam napustio Epl, vratio sam se predavanju, ovaj put u Danekju. Bio sam se upravo oženio i još sam bio na medenom mesecu kada mi je na um pala ideja. Shvatio sam da je ono što sam radio sve te godine bilo pogrešno. Pokušavao sam da napravim bolji kompjuter, a upošte nisam želeo kompjuter! Želeo sam nešto što radi kao sredstvo za primenu. Izgledalo mi je da je moja ideja suviše dobra da je ne podelim sa drugima.

— Znači, to je bila ideja koja vas je inspirisala da oformite novu kompaniju Information Appliance. Kako reč „appliance“ (primena, sredstvo za primenu) označava vaše proizvode?

Da li ste primetili da ne postoje koristanlike grupe vlasnika mašina za pranje veša. Nikom nije potrebna pomoć drugih korisnika da bi upravljao mašinom za pranje rublja. Samo stavite odoču unutra, pritisnete dugme i — odeda je čistal Da bih obeležio obradu informacija, meni ne treba softver i hardver, meni treba neko sredstvo koje će mi izvršiti posao. A koji je to moj posao? Pregledi pokazuju da osamdeset i pet odsto svih vlasnika ličnih kompjutera koristi obradu teksta. Zato mi je potreban tekat procesor — najbolji na svetu. Ali ja sam prilično glup i mogu da zapamtim samo petnaestak komandi. Zato sistem koji je koristim ima samo pet. Mogu da ustanem u tri sata ujutru, odem do kompjutera i samo otkucam ideju koja mi je pala na um.

— Pokušavate da kažete da ste pojednostavili sistem?

Tačno. Pogledajmo kako bi to izgledalo kada bi kompjuterska kompanija proizvodila totere. Probudite se i želite parče tosta za doručak. Prvo što uradite je da uključite toster. Da je toster konstruisala neka obična firma, stavili biste hleb u njega i sve bi bilo gotovo, ali ne, ovaj

„Džobs je rekao da je to luda ideja, da se to neće proizvoditi i da Epl neće da ima ništa sa tim. Pokušao je da obori taj projekt“

toster je konstruisala kompjuterska kompanija! Šta se događa? Kao prvo dvinutina provera tostera. Tada stavite sistemski disk i butujete ga. Posle toga stavite svoj disk za doručak i otkucate Load „Tost“.

Šta se sledeće događa? Pojavljuju se meni. Pita „Koji vrsta hleba želite?“ Ako je to kalifornijski program, nudiće kiflu, domaći hleb, zamlečnik, crni hleb i, na dru, beli hleb. Označeni su A, B, C, D, E. pa zato pritisnete C jer vam se ovog jutra jede baš zamlečnik. Ništa se ne događa, jer ste zaboravili da pritisnete „RETURN“. Možda ste mogli da pomislite da je mašina dovoljno pametna da joj je dovoljno samo C, ali ipak morate da pritisnete RETURN. Da li mislite da će se sada išta dogoditi?

— Paas...?

Naravno da neće. Toster je konstruisala kompjuterska kompanija. Zato on pita: „Jeste li sigurni?“ Sad ste već spremni da ga razupnete o lidu. Da li ste već besni? Zar niste već godinama besni na kompjutere? Ali, zato što ste potrošili par hiljada dolara na njega, podnošite njegovu

U svojoj izuzetno zanimljivoj biblioteci za Prave Programere „Microsoft Press“ je nedavno objavio zbornik tematskih razgovora sa petnaest najbriljantnijih programera današnjice. U toku nekoliko narednih meseci, u okviru serije „Programeri govore“, prenećemo najzanimljivije intervjue iz ove knjige — sa Garljem Kidalom, autorom CP/M operativnog sistema, Džefom Raskinom, vodom projekta „mekintoš“, Džonatanom Saksom, tvorcem „Lotusa 1-2-3“, Bilom Gejtson, tvorcem bejzika i mnogim drugim živim programerskim legendama. Ima li boljih učitelja da nas uvedu u svet programskih ideja, tajni programerskog zanata i tajni uspeha u programerskom poslu?

gluptos, kao i većina ostalog sveta. Milioni ljudi se probijaju kroz ovakve gustosti dok koriste kompjuter.

Zato otkucate „Da“ i pritisnete RETURN, ali dobijate poruku o grešci zato što je trebalo da pritisnete nešto drugo. Gledate u priručnik, ali on vam ne govori ništa, zato što su ljudi koji su ga pisali opisivali prototip koji je promenjen. Konačno, stavite hleb u prorez, naznačite da li želite kafo pečen, normalni ili ozbiljno progoreo i kompjuter vas opita „Da li želite da sačuvate proceduru ovog doručka zato da ne biste morali ponovo da prolazite kroz sve ovo?“. Zato otkucate „Da“ i on vam kaže da ubacite disk u dravaj 1, ali vi tada otkrivате da nemate više formatiranih disketa.

Nazovete svog prodavca i pitate ga da li ima ikakvog načina da formatirate disketu, a da ne izgubite ono što ste napravili tok jutra? On vam odgovara: „Da, samo kupite ovaj hard disk od 3000 dolara sa MS-DOS-om 9.8 i rešite sve vaše probleme. Dobijate ga sa priručnikom i kolicima. Kolicima su za priručnik. „A vi ste već zakazali na posao... Ali, stvari tako stoje. Naš prodavac dosta govori o tom. Predimo na moj kompjuter.“

Raskin seda za „epi II“ opremljen sviftkardom (SwiftCard) i dovoljno oznaka na tastaturi.

Pogledajte ovo. Nema disketa u dravaju i želim da ostavim poruku „Ne zaboravi da doneseš mlieko.“ Uključim ga i počnem da kucam. Nema potrebe za komandama, nema inserta, nema potrebe da se izlazi u editor — mogu odmah da počnem da kucam. Sada želim da ostampam poruku i stavim je u džep, tako da mogu da je koristim kasnije. Pritisnem jednu tugu i već se štampa. Zar to nije zgodno?

**„Tada je Džob preuzeo sve. Jednostaavno je došao i rekao: Ja preuzimam „mekintošev“ hardver“**

— Može li da se pravi tost na ovome?

Ne, mrve upadaju u dravaj.

Šta još mogu korisnici sa ovim spravom?

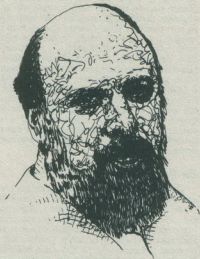
Može lako da se računa. Pre, kada sam koristio teku procesor i želeo da nešto izračunam, morao sam da uzmem džepni kalkulator ili da prizovem program „Sajdkick“ (SideKick); na „meku“ se jednostavno pozove kalkulator. Takođe postoji i mogućnost telekomunikacija.

— Sve u istom programu?

Naravno. Ne postoji bitna razlika između ovih primena. Šta je teku procesor? Koristite ga za pisanje. Šta je telekomunikacioni paket? Koristite ga da pošaljete tekst, ili da primite tekst od nekog drugog. Umešto da dolazi sa tastature, ili izlazi na printer, on se prima ili upućuje preko telefonskih linija. A šta je kalkulator? Koristite ga da operišete sa brojevima. Oni su samo tekst, a rezultat treba da se vrati u vaš tekst. Tako mi je jednog dana sinulo: ako ove stvari rade isto, zašto nemati jedan mali program koji objedinjuje sve te primene?

— Pa, koji ste proizvod razvili da pokriva sva ta područja?

Zove se SwiftCard. To je dodatna ploča za „epi II“. Ne izgleda ništa naročito. Želim da prodajem za samo 89, 95 dolara. Tako je prilično jednostavno da nam se javljaju ljudi koji tvrde da su prodali svoje IBM-ove ili „mekin-



**„Dosta sam naučio o svetu iz rois rojse. Pošto sam otišao iz Epila prodao sam rois“**

toše“ i kupili „epi II“ sa „sviftkardom“ pošto su videli kako radi.

— Da li radi samo na „epu lie“?

„Swiftkard“ radi samo na „epu lie“, za sada. Uskoro ćemo imati identičan softver u proizvodu nazvanom „sviftdisk“ za „epi II“.

— Da li imate nameru da to примените i na drugim kompjuterima?

Mi ništa ne govorimo pre nego što nešto uradimo. Dozvolite da vam pokažem nekoliko prednosti našeg sistema... Ako zaboravite da sačuvate neki tekst pre nego što u računaru učitate novi, on će normalno biti izgubljen. Ovi sistem je dovoljno pametan da se ne dogodi. Ali, evo nečeg još - drastičnijeg. (Raskin ustaje od svog kompjutera i stavlja disketu na džinovski magnet koji visi na zidu) Ovo je sedla potpuno obrisana disketa. Umeću je u dravaj i otkucati nešto teksta i zatim upotrebiti DISK komandu (disk se vrti 4 sekunde). Sad ću uraditi nešto još gore — uključuju kompjuter. Ipak, garantujem vam da, će tekst kad uključim ponovo kompjuter, biti sačuvan. (Uključuje mašinu; za četiri sekunde tekst se pojavljuje na ekranu.) Vidite! Sistem je sačuvao tekst na sasvim neformatiranoj praznoj disketi za samo četiri sekunde. I ne samo to i kurzor je na istom mestu gde smo ga ostavili.

— Najveći broj korisnika je impresioniran mišem. Prihvaćen je na tržištu. Zašto se vama ne sviđa?

Mrzim miševale. Miš zahteva dodatne pokrete rukom koji uporavaju rad. Nisam želeo miš ni na „mekintošu“, ali je Džob insistirao. U ono vreme išlo je sve što je on zahtevao, bilo da dobra ideja ili ne.

— Da li je vaša motivacije za ovaj projekat bila ista kao kada ste pravili „mekintoša“?

Jedan deo motivacije je bio isti, ostali deo je bio izdat. Kod „mekintoša“ sam pokušavao da napravim najbolji kompjuter, koliko je to bilo u mojoj moći. Kada sam stvarao „Inforjeme Acolajans“, nisam više želeo da pravim kompjutere. Samo sam hteo da učinim da dobroti kompjuterske tehnologije budu dostupne svima.

— Vaš izraziti akademski „begrgrund“ mora da je puno vredno u Eplu...?

Tokom Eplovih prvih godina, ja sam bila jedina osoba u čitavoj kompaniji koja je imala fakultetsku diplomu u kompjuterskim naukama. To, međutim, nisam govorio ljudima. Da su znali da sam bio profesor i direktor kompjuterskog centra, možda me ne bi pustili u kompaniju.

— Zašto?

Zato što je postojao izraženi anti-akademski stav u prvim godinama Epla. Mogao sam da dođem i počnem da pišem dobre priručnike zato što sam rekao Zov: „Ti si stručnjak za hardver“, ili Rendiju: „Ti si stručnjak za softver.“ A onda bih im rekao: „E, ja ne znam mnogo o tome. Ja samo znam kako se piše. Ako vi budete pustili mene na miru, puštaju i ja vas.“

— Da li vam smetaju agresivni marketing i napumpavanje? Da li mislite da je to kompjuterskoj industriji pravilo i probleme?

Naravno da jeste. U jednom časopisu sam video kolko oglas preko čitave strane za LCD ekran za „epi lie“ pa sam nazvao tu kompaniju. Rekli su mi da očekuju da se pojavi za deset meseci! Oglaš u koloru preko čitave strane! Mislim da utuđo troše svoj novac.

— Govoreći kao analj koji je video batinu sa obe kraja, šta mislite o odnosu između male i velike kompanije?

Trebalo je da čujete šta je sve Džob nameravao da uradi kada Epi postane veliki. Veoma brzo je zaborvio svoje ideale. Voz se svojih držao mnogo bolje. Jedan od reših ljudi u Eplu koji je zaradio nešto para a nije se mnogo promenio je Brian Hauerd. On je sjajna osoba. Ne znam tačno koliko je bogat, ali on još uvek vozi ista kola koja sam prodao pre deset godina za 75 dolara.

— Da li mislite da će kompjuteri u doglednoj budućnosti raditi i druge stvari osim obrade teksta, spređivota i sličnih poslova? Šta je sa onim obećanjima da će kompjuter kontrolisati čitavu kuću?

Nikad nisam preterano verovao u to, a reći ću vam i zašto. Kao prvo, postojaše namenski sistemi, koji će u stvari sadržavati i mikrokompjuterski i koji će obavljati određeni posao po kući. Takvi uređaji već postoje, ali vi ne mislite o njima kao o kompjuterima; oni su kućni aparati. Kada koristite programabilnu pećnicu, da li razmišljate o činjernici da ona u sebi ima mali mikrokompjuter sa nešto ROM-om i RAM-om i da može da izvršava programe? Da li ih kompišnja na to? Samo stavite hrabu unutra, pritisnete nekoliko dugmad i imate topao obrok za doručak. To je skriveni mikro-projektor. Ima dosta takvih. Da li će kompjuteri raditi to? Nikako. Kao prvo, da li ste videli bilo koja dva kompjuterska proizvoda koja su zalista kompatibilni? Da li mislite da će kompanija XYZ koja proizvodi automatske elektronske prozore zalista napraviti svoj proizvod kompatibilnim sa kompjuterskim kompanije G? Znate i sami kakav je uspeh naterati printer neke kompanije da prodi sa vašim kompjuterom. To se neće dogoditi, to u vezi sa čitavom kućom. Ni na koji način.

— Šta, po vama, programi veštačke inteligencije mogu da doprinesu društvu?

Veštačka inteligencija nas uči dosta o nama samima i o znanju. Bilo koji razum „veštačko-

inteligentni" program neće moći da radi na jeftinim mašinama, bar ne još.

Prava veštačka inteligencija je nešto kao religija. Ljudi su nekada govorili da su odmah iznad neba raj i anđeli. Onda su rakete stigle do tamo i znamo da to nije istina. Teorija je onda promenjena. Tako je i ovde. Čim se nešto postigne, to više nije veštačka inteligencija.

Nekad se mislilo da programi za igranje šaha spadaju u veštačku inteligenciju. Kada sam ja bio student, mogao se dobiti doktorat iz veštačke inteligencije na programu za igranje šaha. Danas možete kupiti elektronski šah za samo 29,95 dolara i niko to više ne zove veštačkom inteligencijom. To je samo mali algoritam koji igra šah.

Kao prvo, postoji problem definicije. A onda postaje sve komplikovano. Ljudi kažu da bi programi morali da razumeju prirodni jezik, ali naša mislija suviše neprecizni da bi kompjuter ili bilo ko drugi mogao da shvati šta treba da se radi; zato i imamo programske jezike. Ako je iko ikada pisao program po pripremi urađenoj na engleskom, zna da se program ne može napisati direktno sa toga, a ako ljudi ne mogu, to se ne može očekivati ni od mašine. Kada se bevite takozvani programima veštačke inteligencije, kompjuter mora da nešto rečnik. Recimo da imate pet komandi i želite da program razume bilo koji ekvivalent tri reči na engleskom: noko bi možda rekao: „Da, mi broj tog radnika“, dok bi drugi kazao: „Da li bi mi, molim te, dao brojčanu oznaku tog radnika.“ To je upravo ono što ljudi koji se bave veštačkom inteligencijom pokušavaju da reše.

Dosta obećanja vezanih za veštačku inteligenciju je pogrešno protumačeno. Ono što nas je veštačka inteligencija već naučila o prirodni jezika je divno. Znači, mislim li da veštačka inteligencija vredni? Apсолutno. Da li mislim da će izbaciti na tržište velike stvari? Nekoliko. Da li mislim da će ispuniti obećanja o kojima čitate u popularnoj štampi? Nimalo. Da li ću ulagati novac u veštačku inteligenciju? Nel.

*Pošto ste postigli uspeh kakve su vam promene bile nametnute?*

Da vam ispričam jednu smešnu priču. Nekad sam posedovao stari narandžasti kamion koji sam bio kupio još 1970. Bio je prešao čitavu Ameriku, moglo je sve da stane u njega, a pozami se moglo i spavati. Divan kamion. A eto, tu sam bio ja, jedan od rukovodilaca u Epiu, i izvodio sam ljude na ručak u tom starom kamionu. Bio sam suviše zauzet da bih pratio kako se kreće moda među automobilima. Konačno, počeo sam da dobijam diskretne sugestije: „Zašto ne nabaviti dobra kola?“. U redu, ja sam odgovoran radnik, ništa što bi štetilo imidžu Epla.

Svi su se vozili unakoliko u „poršeima 928“ ili u „mercedesima“, a ja, jednostavno, nisam želeo luksuzna kola. Ipak, oni su želeli da imam nešto za poštovanje. Tako smo se moj brat i ja zamislili i njemu je pala na pamet ideja. Rečeno je: „Za otprilike iste pare koliko košta novo „mercedes“ ili „porše“ možete da kupite polovnog „rols rojasa“ u izvanrednom stanju. Tako niko neće moći da se žali da nemaš odgovarajuća kola, a opet neće uraditi ono što očekuju od tebe.“ Tako sam kupio „rolsa“. Zaista je čudno. Neki ljudi nikad koji me pre nisu ni primećivali sada su bili voljni da poručaju sa mnom.

U svakom slučaju, otkrio sam koliko malo demokratije imamo. Na benzinskim pumpama ojednom su mi jako pažljivo brisali staklo i nazivali me gospodinom. Ja bih parkirao ispred restorana u zoni u kojoj je zabranjeno parkiranje, a oni bi me nazivali gospodinom, zahvalili se i rekli da im je drago što sam parkirao baš ispred njihovog ulaza, tako da svako može da vidi da vlasnici „rolseva“ jedu kod njih.

Otišao bih do velikih firmi, kao što je Kajzer, gde imaju sve one sline stražare, i samo bih se provozao. Oduvek sam želeo da se provozam unutra, ali nikad nisam mogao da dobijem dozvolu. U „rolsu“ nisam bih se dovezao i rekao: „Imam zakazan sastanak sa gospodinom Mamfordom.“ Samo bi me zahvalili i pustili me unutra. Pošto sam otišao iz Epla, prodao sam „rolsa“.

# Organizacija računara

## IBM PC BIOS

# PC usluga na niskom nivou

Trivijalan zadatak ispisivanja jednog znaka na ekranu može sa na PC računarnima obaviti na više načina. Najpreporučljiviji je, svakako, poziv odgovarajuće DOS funkcije za ovu namenu koja, pored ostalog, vodi i računa o redirekciji izlaza (o kojoj je bilo reči u tekstu PC BUKVAR „...Računari 18“), detektovanju CTRL C (Break) tastera, automatskom pomeranju kursora, i obradi nekih kontrolnih karaktera. Ako je pri tome u CONFIG.SYS detoteci instalisan i ANSI.SYS drajver za konzolu, istom funkcijom će biti moguće, ispisom odgovarajućih kontrolnih sekvenci, proizvoljno postavljati kursor, brisati ekran, postavljati boju, pa čak i predefinisati tastaturu dodeljujući tasterima proizvoljne sadržaje.

### Za ljubitelje brzinske vožnje

Ovakvo široke mogućnosti ne mogu biti ostvarene bez gubitka na nečemu drugom — kod DOS-a je stradala brzina. Sigurno ste primetili da veliki broj programa crta raznorazne ramove i popunjava čitave prozore tekstom velikom brzinom. Ovakva brzina može biti ostvarena na samo dva načina: direktnom komunikacijom sa hardverom, što je postupak koji se primenjuje samo u krajnjim zahtevima, i upotrebom BIOS-funkcija.

Sve funkcije BIOS-a se pozivaju instrukcijom INT uz navođenje broja funkcije iz seta za taj interapt. Opšti oblik bi se mogao definisati sa:

**MOV AH, BROJ FUNKCIJE**  
**MOV ostali registri, parametri**  
**INT broj interapta**

Važi pravilo da sadržaj svih registara koji je vraćaju neku vrednost ostaje nepromenjena. Ono se odnosi i na sve segmentne registre, dok su izuzeci SI, DI i BP, čija promena nije dokumentovana i razlikuje se od funkcije do funkcije. U svakom slučaju, treba računati da njihova ulazna vrednost neće biti sačuvana.

Funkcije su grupisane prema nameni. INT 10h je zadužen za video adapter, a u sebi sadrži 16 funkcija. Jedna od njih je i proizvoljno postavljanje kursora na ekranu, pri čemu tipična procedura za njeno korišćenje izgleda ovako:

**MOV AH,2** : funkcija broj 2 — postavi kursor  
**MOV BH,0**  
**MOV DH,10** : ulazni parametri  
**MOV DL,0**  
**INT 10h** : poziv BIOS-a

U tablici 1 je dat pregled svih značajnijih interapt rutina BIOS-a sa kratkim opisim njihovih namena. U tablici je dato i razgrančanje prema izvoru interapta (S i H). Naime, rutine se oštiro dele u dve grupe. Prva kojoj pripada i video interapt 10h se poziva proizvoljno na zahtev programa. Druga gr-

pa, u stvari, predstavlja rutine za obradu pojedinih događaja u računaru i poziva je sam hardver generisanjem prekida. Zbog arhitekture mikroprocesora 8088, koji ne pravi principijelnu razliku da li je prekid izazvan INT instrukcijom ili je izveden signalom na kontrolnim linija, može nastati mala zabuna. Na primer, interapt broj 9 generiše tastaturu kada je pritisnut neki taster. Rutina za njegovu obradu čita stanje ulazno/izlaznog porta, detektuje o kom tasteru se radi na osnovu njegovog sken kodu itd. Iako se isti prekid može izazvati instrukcijom INT 9, ovakav poziv je u kontekstu bez smisla.

### Pregled interapt rutina BIOS-a:

HEX — NAMENA	IZVOR
5 — ŠTAMPANJE SLIKE SA EKRANA	S
8 — TAJMER	H
9 — TASTURA	H
B — SERIJSKI INTERFEJS 1	H
C — SERIJSKI INTERFEJS 2	H
E — DISK KONTROLER	H
10 — VIDEO	S
11 — INFORMACIJA O KONFIGURACIJI	S
12 — INFORMACIJA O KOLIČINI MEMORIJE	S
13 — DISK	S
14 — SERIJSKI INTERFEJS	S
16 — TASTATURA	S
17 — PARALELNI INTERFEJS	S
18 — ADRESA ULAZA U BEŽIKR ROOM	S
19 — INICIJAL. UPIS SA DISKA (BOOT)	S
1A — ČASOVNIK REALNOG VREMENA	S
1B — INTER. NA CTRL BREAK TASTER	H
1C — TAJMER — SLOBODNA KORISNIKE	H

Izvor S — softver H-hardver

Njihovi brojevi su ipak navedeni u ovoj tablici, jer poznavanje omogućuje izradu posebnih vrsta programa, zamena za BIOS-rutine kojima se mogu postići neki veoma efektni ciljevi. Jedan od primera je interapt 1C, čiji je izvor, doduše, programski, ali se izvodi unutar hardverskog prekida broj 8, tako da ipak spada u H grupu. Ovakvi interapti generiše tajmer za održavanje časovnika realnog vremena, pa se jednostavno izmenom vektora za interapt 1C (koji inicijalno pokazuje na IRET instrukciju) može obezbediti da vaša rutina bude pozvana u regularnim vremenskim intervalima, što obavlja mnoge primene. O ovim tehnikama će biti više reči u nekim od sledećih brojeva „Računara“.

U tablici 2 je dat pregled gotovo svih



**BASIC INPUT OUTPUT SISTEM** predstavlja osnovni program svakog PC računara. Svojim nevelikim mogućnostima, koje su smeštene u manje od 8 K mašinskog koda, u stanju je da obavi samo minimalni ali presudno važan deo posla — direktno obraćanje hardveru računara. Koristeći BIOS kao osnovu, DOS nadograđuje čitavu organizaciju računara. Obracanjem DOS-u u gubl se svaki direktan kontakt sa hardverom i programi postaju lako prenosivi na razne računare. Pa ipak, ni jedan od jačih programa za PC nije odoleo da ne „preskoči“ DOS i iskoristi usluge niskog nivoa BIOS-a. Zašto?

```

// MED - MINI EKRANSKI EDITOR //
// Z.Zivotic(06) "RACUNARI" //
PROMPT equ >
SPACE equ ' '
CRSR_UP equ 'H'
CRSR_DOWN equ 'P'
CRSR_LEFT equ 'K'
CRSR_RIGHT equ 'M'
CRSR_HOME equ 'G'
NEW_LINE equ 13
INTCALL macro res_ah,int_no
mov ah,res_ah
int int_no
ENEM
VIDEOCALL macro res_ah
mov bx,0
push si
push di
push bp
INTCALL res_ah,10h
pop bp
pop di
pop si
ENEM
code segment
assume cs:code
comm_edit proc far
org 100h
start: jmp init
;-----// OLD INT 33 ADR //
save_org dd ?
;-----// NEW INT 33 //
intr: cmp ah,10
jz new_edit
jmp save_org
new_edit: sti
IRP rp,<ax,bx,cx,si,di,bp,es,dx>
push rp
ENEM
push cs
pop ds
assume ds:code
;-----// READ KEY //
read_key: INTCALL 0,16h
cmp al,0
je extend
;-----// DISPLAY CHARACTER //
push ax
VIDEOCALL 14
pop ax
pop bp
csp ah,NEW_LINE
je finish
jmp read_key
;-----// MOVE CURSOR //
extend: push ax
VIDEOCALL 3
pop ax
csp ah,CRSR_LEFT
jne if_right
jne if_right
je read_key
dec di
jmp move_c
if_right: csp ah,CRSR_RIGHT
jne if_up
push cx
push dx
push bp
push es
push bx
inc bx
mov ax,bp
sub ax,si
mov cx,ax
mov byte ptr es:[bx],al
jcxz ret_only
inc bx
store: inc bc
mov al,edit_place[si]
mov byte ptr es:[bx],al
inc si
loop store
ret_only: inc bx
mov byte ptr es:[bx],al
VIDEOCALL 14
IRP rp,<dx,ds,es,bp,di,si,cx,bx,ax>
pop rp
ENEM
;-----// INIT-CODE //
edit_place db 90 dup (0)
init: mov al,21h
INTCALL 30h,21h
mov word ptr save_org,es
push cs
pop ds
mov dx,offset intr
mov al,21h
INTCALL 25h,21h
mov dx,offset init
int 27h
comm_edit endp
code ends
end start

```



rutina BIOS-a koje se mogu koristiti pozivanjem INT instrukcijom. Uz svaku je dat i set ulaznih parametara i sadržaj registra na izlazu (ako ima neko značenje).

### Mini ekranski editor

Korišćenjem BIOS funkcija ostvaren je program koji zamenjuje DOS funkciju b.o.j 10 (upis stringa) u kojoj su obezbeđene osnovne editne mogućnosti pri unošenju naredbi operativnog sistema (primena F1

do F6 tastera itd.). Nova funkcija omogućava da se naredbe unose na mnogo komforiniji način: kursor se može slobodno pomerati po celom ekranu, mogu se izvoditi i ranije otucane naredbe prostim dovodenjem kursora na liniju na kojoj se nalaze itd. Verzija koja je data je, zbog ograničenog prostora, prilično skromna, jer ne omogućuje insert mod, brisanje linije i slične pogodnosti. Osnovni princip rada se zasniva na uzimanju znakova sa ekranske linije na kojoj se kursor nalazi pri pritisku RET tastera. Kao granica sa leve strane je usvojen znak „>“ kojim obično završava DOS prompt, dok je sa desne strane to poslednji znak u liniji.

Program se prevodi u mašinski kod IBM-ovim MACRO asemblerom na sledeći način:

```

MASM med;
LINK med;
EXE2BIN med.exe med.com
DEL med.exe

```

i startuje sa:

A>med

Nakon toga će biti trajno aktivan do prvog resetovanja računara.

## INTERAPT 10

## AH=0 IZBOR VIDEO REŽIMA

AL=0 — 40x2508 tekst  
 AL=1 — 40x25 KOL tekst  
 AL=2 — 80x25 CB tekst  
 AL=3 — 80x25 KOL tekst  
 AL=4 — 320x200, 4 boje, grafika  
 AL=5 — 320x200, 4 CB nijanse, grafika  
 AL=6 — 640x200, 2 boje, grafika  
 AL=7 — MCA tekst

## AH=1 OBLIK KURSORA

CH=prva linija  
 CL=poslednja linija

## AH=2 POSTAVLJANJE POZICIE KURSORA

BH=video strana  
 DH=y koordinat  
 DL=x koordinat

## AH=3 ČITANJE POZICIE KURSORA

BH=video strana  
 DH=y koordinat  
 DL=x koordinat

## AH=4 ČITANJE POZICIE SVETLOSNJ OLOVKE

AH=0 — svet. olovka ne čita poziciju  
 AH=1 — svet. olovka čita poziciju  
 CH=y koordinatne tačke  
 CX=x koordinatne tačke  
 DH=y koordinatne tačke  
 DH=y koordinatne znaka (red)  
 DL=x koordinatne znaka (kolona)

## AH=5 IZBOR AKTIVNE VIDEO STRANE

AL=video strana

## AH=6 BRISANJE/SKROL PROZORA NAGORE

AL=0 — briše sa ceo prozor  
 AL=broj skrol. linija  
 BH=video atribut  
 CH=y koordinatne gornjeg levog ugla  
 CL=x koordinatne gornjeg levog ugla  
 DH=y koordinatne donjeg desnog ugla  
 DL=x koordinatne donjeg desnog ugla

## AH=7 BRISANJE/SKROL PROZORA NADOLE

Igo kao kod AH=6

## AH=8 ČITANJE ATRIBUTA I KARAKTERA NA POZICIJI KURSORA

BH=video strana

AH=atribut  
 AL=ASCII kod

## AH=9 ISPIS KARAKTERA SA ATRIBUTOM NA POZICIJI KURSORA

AL=ASCII kod

BH=video strana

BL=video atribut

CX=broj ponavljanja

## AH=A ISPIS KARAKTERA NA POZ. KURSORA

AL=ASCII kod

BH=video strana

BL=boja za graf. mod (bit 7=1 XOR upis)

CX=broj ponavljanja

## AH=B IZBOR KOLOR PALETE I BOJE

BH=broj palete

BL=broj boje iz palete

## AH=C ČITANJE TAČKE

AL=0—3 broj tačke za 80x200

0—1 broj tačke za 640x200

bit 7=1 — XOR crtanje

CX=x koordinat

DX=y koordinat

## AH=D ČITANJE STANJA TAČKE

CX=x koordinat

DX=y koordinat

## AH=E STANDARDAN ISPIS ZNAKA SA POMERANJEM KURSORA

(prepoznaje BEL. BS. CR. LF kodove)

AL=ASCII kod

BH=video strana (tekst mod)

BL=boja pozadine (graf mod)

## AH=F ČITANJE AKTIVNOG VIDEO REŽIMA

AH=kolonisti format  
 AL=video režim (kao kod AH=0)  
 BH=aktivna video strana

## INTERAPT 11

## AH=0 RESETOVANJE FLOPI DISK KONTROLERA

## AH=1 STATUS FLOPI KONTROLERA

AH=status bit=1 — > greška  
 bit 7 — istek vremena  
 bit 6 — pristup sektoru  
 bit 5 — kontroler  
 bit 4 — čitanje  
 bit 3 — DMA  
 bit 2 — sektor nije nađen  
 bit 1 — disk zaštićen od pisanja  
 bit 0 — pogrešni naredba

CARRY=1 AH=status — greška  
 (status kao kod AH=1)

CARRY=0 AH=0 — uspešno  
 AL=broj pročitanih sektora

## AH=2 ČITANJE SA DISKETE

AL=broj sektora (1—8)

ES:BX=adresa bafera za podatke

CH=trag (0—39)

CL=sektor (1—8)

DI=glava (0—1)

DL=disk jedinica (0—3)

## AH=3 ZAPISIVANJE NA DISKETU

(bit ulaz i izlaz kao kod AH=2)

## AH=4 PROVERA ADRSNIH POLJA NA DISKETI

AL=broj sektora (1—9)

CH=trag (0—39)

CL=sektor (1—8)

DH=glava (0—1)

DL=disk jedinica (0—3)

## AH=5 FORMIRANJE TRAGA

ES:BX — adresa parametar tablice

1 — traga

2 — glave

3 — sektora

4 — format 0=126 bajtova

1=256 bajtova

2=512 ...

## INTERAPT 14

## AH=6 INICIJALIZACIJA SERUSKOG INTERFEJSA

DX=interfejs broj (1=COM1, 2=COM2)

AL=parametar inicijalizacije

bit 765=brzina prenosa

000 — 110 bauds

001 — 150

010 — 300

011 — 600

100 — 1200

101 — 2400

110 — 4800

111 — 9600

bit 43=paritet

10 — bez pariteta

01 — paran

11 — neparan

bit 2=stop bajtova

0 — 1 bit

1 — 2 bita

bit 10=dužina reči

10 — 7 bita

11 — 8 bita

## AH=7 SLANJE KARAKTERA PREKO SERUSKOG INTERFEJSA

AL=ASCII kod

DX=interfejs broj

AH=status

bit 7 AH=0 — znak poslat

bit 7 AH=1 — greška

bit 6—0 — status

(kao kod AH=0)

## AH=8 PRILEM KARAKTERA ZA SERUSKOG INTERFEJSA

DX=interfejs broj

bit 7 AH=0 — prijem u redu

AL=ASCII kod

bit 7 AH=1 — greška u prijemu

bit 6—0 — status

(kao kod AH=0)

## AH=9 STATUS SERUSKOG INTERFEJSA

DX=interfejs broj

AH=status

(kao kod AH=0)

## INTERAPT 16

## AH=0 ČITANJE TASTATURE (SA ČEKANJEM)

## AH=1 ČITANJE TASTATURE (BEZ ČEKANJA)

(ZERO=1 — nema pritisnutih tastera

ZERO=0 — AH=skan kod

AL=ASCII kod

## AH=2 STANJE ŠIFT TASTERA

AL=stanje bit=1 — pritisnut

bit 7 — ins mod

bit 6 — Caps Lock

bit 5 — Num Lock

bit 4 — Scroll Lock

bit 3 — Alt

bit 2 — Ctrl

bit 1 — levi Shift

bit 0 — desni Shift

## INTERAPT 17

## AH=0 ISPIS ZNAKA NA PARELELNI

AL=ASCII kod

DX=interfejs broj (0=LPT1, 1=LPT2)

## INTERFEJS

AH=status

bit 7 — printer slobodan

bit 6 — potvrđen prijem

bit 5 — nema papira

bit 4 — printer on-line

bit 3 — I/O greška

bit 2,1 — se ne koriste

bit 0 — istek vremena

## AH=1 INICIJALIZACIJA PARELELNOG INTERFEJSA

DX=interfejs broj

AH=status

(kao kod AH=0)

## AH=2 STATUS PARELELNOG INTERFEJSA

DX=interfejs broj

AH=status

(kao kod AH=0)

IBM PC

## Bagovi u plavom

Koliko puta imamo prilike da čujemo ponosne vlasnike personalaca u stilu „Ja sam profesionalac, ja radim na IBM PC...“? I ja imam jednog takvog poznanika. Da li ste se ikada zapitali: koliko su sastavljači programa za personalne računare profesionalci, koliko se trude da izbegnu početničke programerske greške, da li se smešno pouzdaju u njihov rad?

Ovde je reč samo o jednoj naredbi sadrženoj u ROMu računara IBM PC/XT koja obično radi dobro, ali ponekad uo krajnje neprijatno da iznenadi. To je naredba PRINT USING. Veoma korisna naredba. Ona omogućava da krivudave redove tabela u ikadiz uredimo da izgledaju lepo kao u — fortranu.

Ako želimo da vrednost nekog broja X predstavimo napr. sa 4 cifre pre decimalne tačke i 5 cifara posle decimalne tačke dovoljno je napisati naredbu

```
PRINT USING „###.###“
```

i naš računar ugledne firme IBM će dobro predstaviti broj X, naravno koliko to vrednost broja X dozvoljava. Ako je

The IBM Personal Computer Basic  
Version 42.00 Copyright IBM Corp. 1981, 1982, 1983  
61981 Bytes free

```
04
FOR I=-6 TO 2:PRINT USING "###":I::
PRINT USING "####.#####":I*11::
PRINT USING "####.#####":I*11*NDT
-6 0.00001000 0.00001000
-5 0.00010000 0.00010000
-4 0.00100000 0.00100000
-3 0.01000000 0.01000000
-2 0.01000000 0.01000000
-1 0.10000000 0.10000000
0 1.00000000 1.00000000
1 10.00000000 10.00000000
2 100.00000000 100.00000000
```

predviđeni broj cifara premlan za broj X računar će ipak — uz komentar — prikazati broj. To je sve u redu, tako bi j trebalo da bude.

Problem nastaje kada je željeni broj decimalnih mesta neki tako neobičan broj kao što je S (i slovima: devet). Nije mi poznato da negde stoji upozorenje da se ne sme štampati broj sa devet decimala. Ako se pokuša dobija se — to je za neverovati! — deset puta manji broj. U memoriji računara broj je predstavljen dobro (koliko konačna dužina mantise dozvoljava), ali ako se zatraži njegova vrednost preko naredbe PRINT USING — onda procenti postaju — promiili!

To je toliko neverovatna i gruba greška da je neophodno snimkom monitora dokumentovati njeno postojanje.

Vratimo se našem poznaniku koji naivno veruje da je profesionalac, čak toliko u to veruje da u to pokušava i druga da ubedi. On takođe veruje i da su programi koji se isporučuju sa računarem besprekorni, veruje da su računari sve inteligentniji. Tim praznovericama treba stati na put — naravno u interesu baš tog našeg poznanika.

Ovde je bilo reči o samo jednoj pogrešnoj naredbi na poznatom računaru. Bilo je dovoljno vremena da se ta greška ispravi. Zar je moguće da ona nekolicom godina nije primećena? Zašto nije ispravljena? Da li se dovoljno razmišlja o rezultatima ako takva greška promiče?

Na kraju, da li smo zaista tako veliki optimisti da verujemo da je to jedina greška u osnovnom softveru računara IBM PC/XT i da li je postojanje neprofesionalnog softvera karakteristika samo vodeće računarske firme?

D. Slavić

# UGRAĐI SAM A<sup>2</sup>

## (elektronsko paljenje za auto)

A2 je elektronsko paljenje za sve benzinske motore — bez obzira na broj cilindara. U A2 su ugrađeni najkvalitetniji kondenzatori, diode, tranzistori, integralna kola...

Elektronsko paljenje A2 ima najbolje sve karakteristike TIRISTORSKOJ i TRANZISTORSKOJ paljenja i u potpunosti zadovoljava najstrožije kriterijume za elektronsko paljenje (startovanje) motora. Elektronsko paljenje A2 omogućava:

- Uštedu u potrošnji goriva (oko 10%). Kod novijih motora sa siromašnijom smesom benzina uštede u potrošnji su veće od 10%.
- Potpunije sagorevanje—manje CO<sub>2</sub> u izduvnim gasovima.
- Mirniji rad motora.
- Bolje paljenje (startovanje) motora, posebno u zimskom periodu.
- Eksploatacioni vek platinskih dugmad višestruko se produžava (preko 100.000 kilometara).
- Posebno PODEŠAVANJE (štelovanja) platina NIJE POTREBNO, jer elektronski sklop A2 u potpunosti preuzima funkciju platina.
- A2 omogućava optimalan rad motora i minimalnu potrošnju goriva.

— Eksploatacioni vek elektronskog paljenja A2 je NEOGRANIČEN. — Jačina iskre (varnice) je 4—5 veća sa elektronskim paljenjem A2, i toliko puta se povećava mogućnost uspešnog startovanja motora. — Za razliku od bezkontaktnog paljenja kod A2, u slučaju kvara, jednostavnim skidanjem kontakta (sa platina i bobine) i povezivanjem kao pre ugradnje nastavljate vožnju.

**Sa ugrađenim elektronskim paljenjem A2 pri ukupnoj potrošnji od 100 litara prosečna ušteda je 10 litara benzina**



- Ugradnja (povezivanje) A2 je brza i jednostavna. Svaki vozac sa minimalnim poznavanjem auto-elektrike, ako postupi prema priloženoj shemi, može da ugradi A2 u svoj auto za 20 minuta
- + GARANCIJA ZA A2 JE GODINU DANA
- + ROK ISPORUKE 15—20 DANA
- + USLOVI IZ NARUĐBENICE VAŽE 30 DANA.

### NARUĐBENICA — RAČ. 24

NEOPOLZIVO naručevanje komada elektronskog paljenja A2 po ceni pri 19.000 din. Navedeni iznos uveden za poštanke troškove PLIČUJEM — prilikom preuzimanja pošiljke. Uslovi iz naruđbenice važe 60 dana.

Prazine i ime \_\_\_\_\_  
Pošt. broj i mesto \_\_\_\_\_  
Ulica i broj \_\_\_\_\_  
Zanimanje \_\_\_\_\_  
telefon \_\_\_\_\_

Štampanim slovima popunite naruđbenicu, zalepite je na dopisnicu i pošaljite na adresu: BIGZ-Agencija DUGA, 11000 Beograd, Bulevar vojvode Mišića 17/III, sa obaveznom naznakom „A2“

cena: 19.000 din.



# humani asembleri

Savremeni trend u razvoju mikroprocesora nalaze takozvanu registarski orijentisanu organizaciju, to jest krajnje redukovan pristup memoriji i minimalno korišćenje apsolutnog adresiranja, pa i H-asembler sledi takav pristup. To može stvoriti izvesne poteškoće pri „humanizovanju“ nekih za starijih procesora koji nisu registarski orijentisani (6502, na primer).

Na slici 1 je data takozvana tablica preslikavanja standardnog i H-asemblera za Z80. Ova tablica je od presudnog značaja za pisanje „humanizatora“, budući da sadrži sve neophodne elemente za prevodenje mnemoničkih oznaka. Vidimo da je „humanizacija“ prošla prilično „glatko“, to jest da nije rezultirala u značajnom povećanju dužina mnemonika. Primetno su „narasili“ mahom oni mnemonici koji označavaju ređe korišćene operacije. Procesor Z80 je izuzetno registarski orijentisan, što pogoduje lakom uklapanju H-asemblera.

Drugi procesor po popularnosti je potpuna suprotnost Z80. Striktno je memorijski orijentisan, a registre koristi samo za ono što se nikako drugačije ne može uraditi. Radi se, naravno, o 6502, čija tablica preslikavanja je data na slici 2. Na ovoj tablici se ne vide nikakvi naročiti problemi, pa izgleda da i ovaj procesor može „leže“ na H-asembler. Razlog tome je što ova tablica nije razrađena po adresnim modovima (kao što je slučaj sa adresnim modovima Z80) već su oni dati izdvojeno na slici 3, sa koje vidimo da takozvano memorijski indirektno adresiranje rezultira vrlo rogalobnom konstrukcijom (nije baš ni svakom procesoru sudeno da bude „human“).

Postoji nekoliko metoda kojima se može implementirati H-asembler. Prvi je pisanje kompletnog asemblera sa novim mnemoničkim, što je najdosledniji ali i najduži put kojim polaze samo firme poput Hewlett-Packard-a pri razvoju softvera za procesore

koji su već dizajnirani kao „humani“ (procesor računara HP-71B, čiji vlasnici raspolažu takvim „dosiedno-humanim asemblerom“). Drugi pristup omogućava korisniku da istovremeno radi i sa starijim i sa novim mnemoničkim, to jest da ih „meša“ po slobodnom izboru. To se postiže posebnim programom nazvanim

## Preprocesor

čija je glavna funkcija (grubo rečeno) da svaki string karaktera na ulazu koji postoji u tablici preslikavanja zameni odgovarajućim stringom iz iste tablice. Ova metoda nije, naravno, izmišljena radi implementacije H-asemblera, već je to jedna odšta metoda, korišćena za prevodenje nekog programskog jezika uz minimalan utrošak memorije, koja je svoju punu afirmaciju doživela pojavom jezika C (naredbe DEFINE i INCLUDE u stvari samo obezbeđuju tablicu preslikavanja za preprocesor).

r=#	LD	r,S				r=#	b	BIT	b,r	
r=#	LD	r,n	PORT.C=DAT.HL	I	OUTI	?DAT.HL=b	b	BIT	b,(HL)	
r=DAT.HL	LD	r,(HL)	PORT.C=DAT.HL	IR	OTIR	?DAT.(XY+d)=b	b	BIT	b,(XY+d)	
r=DAT.(XY+d)	LD	r,(XY+d)	PORT.C=DAT.HL	S	OUTD	r=#	b	SET	b,r	
DAT.HL=r	LD	(HL),r	PORT.C=DAT.HL	DR	OTDR	DAT.HL=b	b	SET	b,(HL)	
DAT.(XY+d)=r	LD	(XY+d),r	A=PORT.C		IN	A,(C)	b	SET	b,(XY+d)	
DAT.HL=r	LD	(HL),n	A=PORT		IN	A,(C)	b	RES	b,r	
DAT.(XY+d)=n	LD	(XY+d),n	A=PORT	n	IN	A,n	b	RES	b,(HL)	
r,BIB,C,D,E,H,L,A			PORT=A	n	OUT	A,n		RES	b,(XY+d)	
A=DAT.BC	LD	A,(BC)	A=A+S	ADD	A,S	ROT.C.circ	S	RL	S	
A=DAT.DE	LD	A,(DE)	A=A+CY+S	ADC	A,S	ROT.S	S	RLC	S	
A=DAT.	LD	A,(nn)	A=A-S	SUB	A,S	ROT.>.circ	S	RR	S	
DAT.BC=A	LD	(BC),A	A=A-CY+S	SBC	A,S	SHIFT.<	S	RRC	S	
DAT.DE=A	LD	(DE),A	A=A+ndS	ADC	S	SHIFT.>.ar1t	S	SLA	S	
DAT.=A	LD	(nn),A	A=A+S	OR	S	SHIFT.>.l0g	S	SRA	S	
A=I	LD	A,I	A=A+S	XOR	S					
A=B	LD	A,B	FLAG=A	CP	S					
I=A	LD	I,A				BIB,C,D,E,H,L,A,DAT.HL,DAT.(XY+d)				
R=A	LD	R,A	BIB,C,D,E,H,L,A,DAT.HL,DAT.(XY+d)			GOTO	label	JR	label	
d=#	LD	d,d,nn	A=A+	n	ADD	n				
XY=	LD	XY,nn	A=A+CY+	n	SUB	n				
d=#DAT.	LD	d,d,(nn)	A=A-CY-	n	ADC	n				
XY=DAT.	LD	XY,(nn)	A=A-CY-	n	AND	n				
DAT.=dd	LD	(dd),dd	A=A+nd	n	AND	n				
DAT.=XY	LD	(nn),XY	A=A+r	n	OR	n				
461BC,DE,HL,SP			A=A+r	n	XOR	n				
XY1X,XY			FLAG=A-	n	CP					
SP=HL	LD	SP,HL	r=#	I	INC	r	IF.=.G0ABS	label	JP	Z,label
SP=XY	LD	SP,XY	DAT.HL=DAT.HL+1		INC	(HL)	IF.=.G0ABS	label	JP	NZ,label
RET=qq	PUSH	qq	DAT.(XY+d)=DAT.(XY+d)+1		INC	(XY+d)	IF.<.G0ABS	label	JP	C,label
RET=xy	PUSH	xy	r=-1	DEC	r	IF.>.G0ABS	label	JP	NC,label	
SP=RET	POP	xy	DAT.HL=DAT.HL-1		DEC	(HL)	IF.odd.G0ABS	label	JP	PO,label
XY=RET	POP	xy	DAT.(XY+d)=DAT.(XY+d)-1		DEC	(XY+d)	IF.even.G0ABS	label	JP	PE,label
461BC,DE,HL,AF			A=DEC	DAA		IF.=.G0ABS	label	JP	P,label	
DE=HL	EX	DE,HL	A=ADD	AD		IF.=.G0ABS	label	JP	N,label	
AF=AF	EX	AF,AF	A=A	NEG		IF.=.G0SUB	label	JP	Z,label	
ALTERNATE	EXX	EX	CY=B	CPL		IF.=.G0SUB	label	JP	NZ,label	
DAT.SP=XY	EX	(SP),XY	CY=I	SCF		IF.<.G0ABS	label	JP	C,label	
DAT.SP=XY	EX	(SP),XY	NOF	NOF		IF.>.G0SUB	label	JP	NC,label	
DAT.DE=DAT.HL	LBI		INT0FF	DI		IF.odd.G0SUB	label	JP	PO,label	
DAT.DE=DAT.HL	LDIR		INTON	DI		IF.even.G0SUB	label	JP	PE,label	
DAT.DE=DAT.HL	LDD		INT=B	IM	0	IF.=.G0SUB	label	JP	P,label	
DAT.DE=DAT.HL	LDR		INT=2	IM	1	IF.=.G0SUB	label	JP	N,label	
7A=DAT.HL	I	CP1	HL=HL+SS	ADD	HL,SS	RETURN		RET		
7A=DAT.HL	IR	CP1R	HL=HL-CY+SS	ADC	HL,SS	INT.RETURN		RETI		
7A=DAT.HL	DR	CP1D	HL=HL-CY-SS	SBC	HL,SS	NONRASC.RETURN		RETN		
7A=DAT.HL	DR	CP1R	XY=XY+dd	ADD	XY,dd	RESTART	d	RET	d	
7A=PORT.C	I	INI	SS=SS-1	INC	SS	IF.B-100.G0TD	label	DJNZ	label	
7A=PORT.C	IR	INIR	XY=XY+1	INC	XY					
7A=PORT.C	D	IND	SS=SS-1	DEC	SS					
7A=PORT.C	DR	INDR	XY=XY-1	DEC	XY					
			861BC,DE,HL,SP							
			461BC,DE,XY,SP							

Ideja o humanom asembleru je nastala, pre svega, iz nekih čisto akademskih diskusija o odnosu čovek-računar i nismo nameravali da je na bilo koji način širimo. Ispostavilo se, međutim, da čitaoci „Računara“ žele i mogu mnogo više, pa ćemo (uz vašu saradnju) pokušati da „humanizujemo“ asemblere za danas najpopularnije procesore. Pri tom, uglavnom, neće biti davan nikakvi gotovi programi tipa „ukucaj-i-startuj“; već opšti algoritmi na osnovu kojih svaki pažljiv čitalac „Računara“ može „humanizovati“ svoj asembler. Ukoliko, međutim, dobijemo od vas kvalitete (i dobro dokumentovane) programe za „humanizovanje“, nećemo žaliti ni prostora ni novca.

Da bi pretprocesor radio, mora postojati neki blok u memoriji koji sadrži izvorni kod (to jest, tekst iskucan preko tastature ili unesen sa spoljašnjeg medijuma) programa. Realizacija ovakvog bloka je izuzetno zavisna od računara (ili barem operativnog sistema), te se njom nećemo detaljnije baviti. Potrebno je za svaki asemblerski program za koji radimo pretprocesor (to jest humanizujemo ga) pogledati gde i kako čuva izvorni kod posle unošenja sa tastature, za vreme prevodjenja i po učitavanju sa spoljnog medijuma. Najsigurniji način je:

- 1 — učitati izvorni kod sa spoljnog medijuma;
- 2 — izvršiti pretprocesorski program;
- 3 — snimiti novi izvorni kod na spoljni medijum;
- 4 — primeniti uobičajenu proceduru asembliranja za izvorni kod na spoljnog medijumu.

Dobri poznavaoči semblerskog programa mogu, međutim, uštedeti dosta vremena tako što će, recimo, preraditi asembler da automatski učitava i startuje pretprocesor tako da za korisnika sve ostaje „po starom“. Pri ovakvom „povezivanju“ pretprocesora treba, međutim, biti vrlo oprezan da ne bi došlo do takozvanog prekrivanja koda, što može imati vrlo neugodne posledice i izazvati „pad računara“.

Osnovni algoritam pretpostavlja da smo locirali blok sa izvornim kodom koji nosi naziv (preko koga mu pristupamo) FILE u kome svakom redu pristupamo primitivnim naredbama next (pristupa sledećem redu koji postaje tekući) i curr (pristupa tekućem redu). Algoritam je dat u pseudo-jeziku uz korišćenje uobičajenih kontrolnih struktura, s tim da strelica označava dodeljivanje vrednosti (:= u paskal varijanti a:= u ostalim jezicima). Osnovni algoritam pretprocesora je:

```
while not end of FILE do
begin
  RED ← next (FILE);
  RED ← translate (RED);
  curr (FILE) ← RED
end
```

end i sastoji se u iščitavanju sukoesivnih redova (čiji sadržaj se upisuje u pomoćni string — RED), prevodenju reda (pozivom procedure translate) i vraćanju prevedenog reda u FILE. Stvari su, dakle, jednostavne da jednostavnije ne mogu biti. Treba samo rešiti pristup bloku u kome je tekst (što pažljivo čitaocu „Računara“ nije veliki problem) i jednu malu (malečku — ovolincnu; „sitnicu“ —

## Prevodjenje

jednog reda, koje je ovde označeno procedurom translate. Pogledajmo kako bi se to uradio takozvanim direktnim pristupom (u nas poznatijim kao „glavom o zid“). Jedno takvo rešenje bilo bi:

## 21/humani asembleri

AA+VC+	M	ADC	M
AND&V	M	AND	M
SHIFT,C	M	ASL	M
IF,<,DOTO	label	BCC	label
IF,=,DOTO	label	BCS	label
IF,≠,DOTO	label	BEQ	label
7&ad	M	BIT	M
IF,<,DOTO	label	BRl	label
IF,=,DOTO	label	BNE	label
IF,≠,DOTO	label	BPL	label
BREAK	label	BRK	label
IF,over,DOTO	label	BVC	label
IF,over,DOTO	label	BVS	label
CL		CLC	
CL		CLD	
TINTON		CLI	
DIVER#		CLV	
FLAG#&X		CMF	
FLAG#&X		CPX	
FLAG#&X		CPY	
IN&V		INC	M
X&X		INX	M
Y&Y		INP	
IN&V		INT	
X&X-1		DEC	M
Y&Y-1		DEY	
GD&B	label	JMP	label
GD&B	label	JSR	label
A&	M	LDA	M
X&	M	LDX	M
Y&	M	LDY	M
SHIFT,+.109		LSR	M
NOP		NOP	
AND&op	OP	ORA	
RET#A		PHA	
RET#P		PHP	
AND#RT		PLA	
P#RET		PLP	
ROT,C		ROL	M
ROT,S		ROR	M
INT.RETURN		RTI	
RETURN		RTS	
AND#CY-		SEC	M
CY#1		SEC	
SET&B		SED	
INT#OFF		SEL	
H#A		STA	M
N#V		STX	M
X#V		STY	M
X#A		TAX	
Y#A		TAY	
X#V		TSX	
AND		TXA	
S#V		TXB	
X#V		TYA	

Substitute M with the appropriate addressing mode.

procedure translate (STRING):

begin

```
(TEMP, FLAG) search (STRING);
IF FLAG then STRING TEMP;
return (STRING)
```

end

Oformili bismo naziv H-mnemonika (op-kodova i modifier-a u svim kombinacijama) sortiran po ASCII poretku, tako da svakom mogućem mnemoniku standarda odgovara tačno jedan element našeg H-niza. To je, u stvari, potpuno razvijena tablica preslikavanja sa svim mogućim operandima. Procedura search traže naš niz binarnim metodom (koji je potanko objašnjen u „Računarima“ broj 21) i ukoliko pronađe STRING, vraća njegovu „zamenu“ iz skupa standardnih mnemonika i FLAG=true, a ukoliko ga ne pronađe vraća prazan string i FLAG=false. Ovaj način bi, čak, bio dosta brz, ali bi „pojeo“ toliko memorije da bi samo hardisk mogao da „namiri“ njegovu „halapljivost“.

Podelimo sada posao na dva dela, tako da prvi deo prevodi samo op-kod a drugi samo operand (ili modifier, kako su stari

„mašinski vukovi“ navikli da govore). Na izgled smo od jednog posla napravili dva, no to je samo privid. Operand, praktično, nije ništa drugo (u H-assembleru) do numerička konstanta ili oznaka koju je potpuno nepotrebno prevoditi (u direktnom pristupu svaka takva konstanta je za sobom vukla 256 ili čak 65536 elemenata niza) pa smo tako, u stvari, sveli posao na pretraživanje niza op-kodova kojih ima mnogo manje (kod 6502 samo 56). Ovakva realizacija bi imala pristojnu brzinu čak i iz bezjaka i zauzimala bi relativno malo memorije. Opšti algoritam za ovaj pristup je:

procedure translate (STRING):

begin

```
TEMP ← optrans (STRING);
TEMP ← TEMP + modtrans (STRING);
return (TEMP)
```

end

Postoji, međutim, jedna nezgodna stvar koja je posledica same filosofije H-asemblera koja pretenduje da ukloni stalno razdvajanje op-koda i operanda te tako učini kod razumljivim. Na taj način (recimo kod indeksnog adresiranja) neke konstante još uvek se nalaze u polju op-koda, a samo jedna je dovoljna za „eksploziju“ našeg niza (sa novih 256 ili 65536 članova). Dalje mnemonik r=# ima tačno 49 mogućnosti za Z80, što je opet previše. Mogli bismo izrazе razdvajati na simbol=ali ovaj simbol nema jedinstvenu interpretaciju, pa bi to samo dalje usložnjavalo i usporavalo algoritam. Dakle,

## Možete (i mora) to i bolje

Ako pažljivo pogledamo tablicu preslikavanja za Z80, videćemo da mnemonika sa indeksnim adresiranjem ima vrlo malo, što znači da ih lako možemo preoznati i posebno obraditi kao i neke specijalne slučajeve (kao već spomenuti r=#). Potom je veliki broj OP-kodova fiksni, to jest nemaju mogućnosti rada na različitim registrima, a i oni koji nisu fiksni imaju najviše 4 do 7 varijacija koje su ili na početku ili na kraju stringa, pa možemo (shodno raspoloživoj memoriji i željenoj brzini) odlučiti da ih sve stavimo u sortirani niz ili da neke od njih obradimo kao posebne slučajeve. Pri tome možemo koristiti sledeći „trik“: bilo koja grupa mnemonika (fiksni, specijalni, slabovarijajući) da je obradena, mora završetkom svoje obrade omogućiti dalje ispitivanje op-koda (da bi se izbegla mogućnost dvostruke interpretacije izlog koda).

Problem oznaka (ili labela) na početku reda rešićemo na standardni način tako što ćemo zahtevati da ili oznaka počinje u drugoj koloni ili sadrži određeni specijalni znak (# na kraju — Z80 asembler, na početku — 6502 asembler) čime ćemo manipulisanje oznakama učiniti trivijalnim, pa ga u konačni algoritam nećemo ni uneti. Postoji, takođe, mogućnost eliminisanja svih blankova pri obradi, čime se, s jedne strane, olakšava rad pretprocesora (kao prepoznava op-kod, zna da na prvom slede-

ABSOLUTE	DAT. nn
#-PAGE	DAT. n
IMMEDIATE	n
ABSOLUTE, X	DAT. (X+nn)
ABSOLUTE, Y	DAT. (Y+nn)
(INDIRECT, X)	DAT. (DAT. (X+nn))
(INDIRECT, Y)	DAT. (DAT. (nn)+Y)
(#-PAGE, X)	DAT# (X+nn)
(#-PAGE, Y)	DAT# (Y+nn)
INDIRECT	DAT. (DAT. (nn))



## Trideset i tri proste sekvence

**Greška urednika ove rubrike koji je, u januarskim „Računarima“, napisao da se rešenja primaju do 1. februara 1986. godine je, na sreću, ostala slabo primećena: u predviđenom smo roku primili rekordnih 329 pisama sa konkretnim rešenjima petnaeste pitalice; pogrešne odgovore nismo brojali, ali ih nije bilo mnogo. Ako se po jutru dan poznaje, evidentiranje rezultata našeg godišnjeg takmičenja neće biti baš mali posao!**

Moram da priznamo da nas je petnaesta pitalica uplašila: hoće li ova rubrika biti ukinuta kao društveno štetna? Mnogi čitaoci „Računara“ su, sa pravom ili ne, zaključili da je rad sa velikim prostim brojevima prekrupan zaloga za „spektrume“ i „komodore“ pa su posetili razne računске centre i angažovali jače mašine. Kada se uzme u obzir da su prosečna rešenja potrošila i po par procesorskih sati, ukupno vreme koje su jugoslovenski računski centri investirali u rešenje beskorisne stvari kao što je ova pitalica dostiže zbir koji se ne usudujemo da pomenemo!

### Počelo je naivno

Šalu na stranu, podsetimo se petnaeste pitalice koja je započinjala sasvim naivno: napisamo neki jednocifren prost broj (2, 3, 5 ili 7). Zatim, sa leve ili desne strane, dopišemo neku različitu cifru tako da rezultat ponovo bude prost i nastavljamo ovaj postupak što duže. U postavi zadatka smo napomenuli da najduža „prosta“ sekvenca ima osam brojeva, jer bi devetocifren broj sastavljen od različitih cifara bio deljiv sa devet, tj. ne bi bio prost.

Malu zabunu unosi nula. Nema je, jasno, smisla dopisati zdesna, jer bi tako dobili n broj bio deljiv sa 10. Dopisivanje nule sleva je, u brojnim sistemima koji koristimo, besmisleno ali ipak dopustivo — na ovaj bi se način dobila sekvenca od devet prostih brojeva, od kojih bi dva bila jednaka. Smatrajući ovakvo rešenje nebitnim (i dalje se sastoji od 8 brojeva), ograničili smo se na sekvence dužine osam.

Definicija kaže da je prirodan broj N prost ukoliko je bez ostatka deljiv isključivo

sa 1 i sa N. Kako da proverimo da li neki broj zadovoljava ovu definiciju? Načina ima više; dele se na strahovito spore i još sporije.

Najgori način je odgovoran za višestruka rešavanja naše pitalice: probamo da li je N deljiv sa 2. Ako nije, probamo da li je deljiv sa 3, 4, 5, ... N-1; ako N nije deljiv ni sa jednim od ovih brojeva, sigurno je prost! Nevolje je, međutim, što je ogroman deo ovakvih proba suvišan: ako broj nije deljiv sa 2, on sigurno nije deljiv ni sa 4, 6, 8... što znači da se najgori algoritam malo poboljšava dodavanjem jednog STEP 2. Većina ispitivanja je i dalje besmislena: ne treba isprobavati delioce do N-1 već samo do SQRT (N). Možda ćemo tako propustiti neki delilac — broj 21 je, na primer, očit deljiv sa 7 pri čemu je 7 veće od SQRT (21)? Ne brite — propuštanje nekog delioca ne utiče na zaključak o tome da li je broj prost: broj 21 je, na primer, deljiv sa sedam ali je deljiv i sa 3 pri čemu je 3 manje od SQRT (21)!

Sumirajući dosadašnju diskusiju, dolazimo do petlje FOR i=3 TO SQRT (N) STEP 2 koja, međutim, i dalje nije baš slavná: proveravaćemo, na primer, da li je broj deljiv sa 9 pošto smo već utvrdili da nije deljiv sa 3! Pravi se algoritam nameće: treba ispitati da li je N deljiv sa prostim brojevima koji su manji od njegovog korena. Nevolja je što te proste brojeve obično ne znamo unapred: obzirom da u našem zadatku treba testirati osmociifrene brojeve, trebalo bi pripremiti tablicu prostih brojeva manjih od SQRT (100.000.000)=10.000, a takvih ima poprilično mnogo — za njihovo generisanje treba utrošiti određeno vreme a za memorisanje određeni prostor. Zato se u praksi obično koristi (najmanje loše) kompromisno rešenje koje ćete upoznati analizirajući proceduru *prost* sa slike 1.

### Jedna prosta formula

I pored ogromnih napora matematičara, do danas nije pronađen opšti oblik prostog broja, tj. formula koja bi, za razne vrednosti nekog parametra, generisala niz prostih brojeva (opšti oblik parnog broja je, na primer, 2<sup>P</sup>). Ovdavno je, međutim, utvrđeno da su svi prosti brojevi veći od 3 oblika 6<sup>P</sup>+1 ili 6<sup>P</sup>-1 (nemoguće nas pogrešno shvatiti: ako je broj prost, on je sigurno oblika 6<sup>P</sup>+1; ako je neki broj oblika 6<sup>P</sup>-1, on može ali ne mora da bude prost) što znači da je za testiranje nekog broja N dovoljno pokušati njegovo deljenje sa 2, 3 i svim brojevima oblika 6<sup>P</sup>+1 i koji su manji od njegovog kvadratnog korena. Ukoliko, dakle, još nekada budete rešavali zadatke u kome se pominju prosti brojevi, setite se „Računara 24“ i procedure *prost*; to je otprilike najmanje loša postojeća varijanta.

Pošto smo naučili da određujemo proste brojeve, vraćamo se problemu koga nije mnogo teško rešiti: počnemo od broja 2 i pokušavamo da mu sleva ili zdesna dopišemo neku cifru, pri čemu s desne strane nema smisla dopisati 2, 4, 5, 6 ili 8. Ispitano da li je rezultat prost i ako nije pokušamo sa nekom drugom cifrom. Vrlo je važno obezbediti takozvani *backtracking* — ako dodavanje cifre nije uspeo, treba se vraćati nazad i pokušavati sa drugim ciframa čak i nekoliko nivoa unazad. Uz malo paskala i malo rekurzija, dolazimo do programa sa slike 1. Iako smo primili i dosta rešenja na bežikju (i bežik, znate, omogućava rekurziju), odlučili smo da za „završić-

ćem karakteru počinje operand — ako postoji), a, s druge, oduzima vreme za eliminaciju i traži čuvanje originalnog stringa (da bi nepromenjen bio vraćen ako u njemu nema humanih mnemonika). Eliminisanje blankova, međutim, oslobađa korisnika stalne napetosti izazvane mogućnošću da nedostatak jednog jedinog blanka onemogućiti prevodenje programa pa naš assembler, budući da je „human“, mora imati u ovoj operaciji. Opšti izgled završnog algoritma bio bi dakle:

### procedure translate (TEXT)

begin

STRING ← eliminate (TEXT)

(TEMP, FLAG) ← special (STRING)

if not FLAG then

begin

(TEMP, FLAG) ← fix (STRING)

if not FLAG then

(TEMP, FLAG) ← lovable (STRING)

end

if FLAG THEN RETURN (TEMP + mods-trans (STRING))

else return (TEXT)

end

gde procedura *eliminate* vrši eliminaciju blankova, procedura *special* obrađuje specijalne slučajeve (indeksno adresiranje, eventualno „zaostale“ konstante u op-kodu i druge „neugodne slučajeve“ koji se od procesora do procesora razlikuju), procedura *fix* koristi algoritam direktnog pristupa budući da radi sa fiksnim op-kodovima (pa ne mora koristiti suviše veliki niz sa tablicom preslikavanja) i, konačno, procedura *lovable* obrađuje op-kodove male promenljivosti koje možemo obrađivati posebnim postupkom (utvrđivanje varijabilnog dela, pa zatim obrada metodom binarnog pretraživanja) ili na isti način kao i *fix* (kada nam ova procedura, zapravo, nije ni potrebna).

Moguće je, naravno, kompletan skup mnemonika „razbiti“ na sitnije klase, no to može rezultirati značajnim padom brzine (identifikacija klase je često mnogo duža od binarnog pretraživanja, koje je najbrži način nalazanja podataka na jednoprocorskim mašinama). Očekujemo, dakle, vaše „humanizujuće“ programe (sa ovim ili možda nekim efikasnijim algoritmom) za „Biblioteku programa“ — na radost svih onih koji tek ulaze u svet mašinskog programiranja.

Moguće pogreške u ulaznim podacima nismo razmatrali pre svega zato što pravimo preprocesor, a on treba što manje da se „bakće“ sa takvim stvarima kako bi dobio na brzini i efikasnosti. Jednu stvar, ipak, mora da proverí. Kada prepozna op-kod i krene u analizu operanda, mora da prekine rad i izda poruku o grešci ukoliko operand nije korektan (ne sadrži ni broj ni oznaku, ili kod inkrementa i dekrementa postoje dva različita operanda itd.). U slučaju da op-kod nije prepoznat, preprocesor vraća originalni red u FILE, tako da će tek assembler eventualno prijaviti neku grešku.

**Zarko Berberski**







## Kombinatorika u praksi

Rokovi za prodaju rukopisa u štampi su, na žalost, takvi da ovaj tekst pišemo samo par dana po izlasku „Računara 23“, što znači da neki od vas tek započinju pisanje programa koje treba da objavimo za dva meseca. U radoznom očekivanju vaših priloga o kojima ćemo pisati u sledećim „Računarima“, pripremili smo novi programerski izlaz.

Pošto je „Programerska radionica“ savim novu rubriku, verujemo da odredben broj čitalaca ovih redova nije upoznat sa našim prethodnim tekstom, pa ćemo ukratko ispričati o čemu se radi. „Programerska radionica“ treba da opremi Prave Programere Pravim Alatkama, a te alkatke nisu samo kompjuteri, štampači, diskovi, modeli, kompjajeri i slične draguljice već i procedure koje će u pravom trenutku iskoristiti kao potprograme. Ukoliko sve te potprograme pišemo kada nam trebaju, stotinu čemo puta ispisivati iste redove, a rezultati će biti vrlo daleko od savršenstva. Zato smo odlučili da, u saradnji sa čitaocima „Računara“, sastavimo odredben broj često potrebnih programa koje ćemo čuvati na kasetama ili diskovima i koristiti kada nam ustreba.

Pošto „Računare“ čitaju vlasnici raznoraznih kompjutera, programi koje ćemo objaviti treba da budu što univerzalniji: ne zanima nas rešavanje problema uz muziku i grafiku. Ne interesuju nas, takođe, ni asemblerske majstorije — treba nam dobar program na jeziku ili paskalu koji će biti svima razumljiv i jednostavan za modifikovanje. Kako ćemo doći do tako dobrog i univerzalnog programa? Opisao ćemo neki problem, dati nekoliko predloga za njegovo rešavanje i sačekati tridesetak dana. Od programa koje dobijamo izdvojićemo najbolji i, uz eventualne prerade, objaviti njegov listing u „Računarima“; ovo će rešenje, jasno, biti honorisano a „Računari“ će iskoristiti priliku da dodele zaslužene nagrade i autorima par programa koji su ušli u uži izbor za objavljivanje.

## Drugi zadatak: kombinatorika

Ako ste ikada pisali program koji se ne bavi samo „žvakanjem brojeva“ ili kontrolom periferijskih uređaja, upoznati ste sa značajem takozvanih „numeričkih metoda“ — primene računara u mnogobrojnim oblastima kao što su sortiranje, pretraživanje, generisanje slučajnih brojeva i, naravno, veštačka inteligencija. Kombinatorika je, verujemo, jedan od kamena temeljaca mnogih oblasti programiranja, pa ćemo je uvrstiti u spisak elementarnih aljati. Verujući da mnogi hakeri nisu baš savršeno upoznati sa razlikama između kombinacija, permutacija i varijacija, posvetićemo sledeće redove elementarnoj kombinatorici.

**Permutacije bez ponavljanja** je najjednostavnije za razumevanje; uzmemo nekoliko različitih elemenata i raspoređujemo ih na sve moguće načine. Elementa A, B i C, na primer, možemo da rasporedimo na šest načina: ABC, ACB, BAC, BCA, CAB i CBA. Ukoliko, uopšte, imamo N različitih elemenata, možemo da ih rasporedimo na  $N! = 1 \cdot 2 \cdot 3 \dots N$  različitih načina.

Jedna od zanimljivih primena permutacija će zauvek ostati u sećanju urednika ove rubrike: prošlog leta je trebalo pregledati rešenja osme Dejanove pitalice, tj. programe koji sortiraju pet skalara u rastući redosled. Najbolji test ovakvog programa je njegovo startovanje za sve moguće rasporede nekih pet skalara, a takvih raspoređa ima  $5! = 120$ . Pošto urednik ove rubrike nikada nije našao vremena da napiše program za generisanje svih permutacija, zadržali smo se da, poput svakog smrtnika, isproba desetak setova ulaznih veličina i tako, pod blaženim delovanjem Marjiljevih zakona, objavi program koji ne radi baš

```

100-Pascal compiler V. 01.89
1  # Function pitalica (input,output):
2  # var niz: array [1..5] of integer;
3  #   p: array [1..5] of boolean;
4  #   d: array [1..5] of integer;
5  #   štampa: boolean;
6  #
7  # Function procit (integer): boolean;
8  # var n: array [1..5] of integer;
9  # begin
10 #   if (n=0) or (n=3)
11 #     then p:=true;
12 #   else
13 #     begin
14 #       if (n mod 2=0) or (n mod 3=0)
15 #         then p:=false;
16 #       else
17 #         begin
18 #           k:=n*(n-1); n:=n; add:=n;
19 #           while (k=0) and (n mod 4=0) do
20 #             begin
21 #               n:=n-4; add:=n+add;
22 #             end;
23 #             p:=true;
24 #           end;
25 #         end;
26 #       end;
27 #     end;
28 #   procedure treci(n: integer);
29 #   var i: integer;
30 #   #   if (d[n] and procit(n))
31 #     then if (d[n] and procit(n))
32 #       then
33 #         begin
34 #           d[n]:=d[n] + 1;
35 #           p[(d[n]-1) * štampa:=true;
36 #           for k:=1 to 5 do
37 #             if p[k] then
38 #               begin
39 #                 treci(n+k);
40 #                 treci(n-k);
41 #                 p[k]:=true;
42 #               end;
43 #             d[n]:=d[n] - 1;
44 #             if (d[n]=0) and štampa then
45 #               begin
46 #                 if (d[n]=0) and štampa then
47 #                   begin
48 #                     štampa:=n;
49 #                   end;
50 #                 if (d[n]=0) and štampa then
51 #                   writeln(n);
52 #                 if (d[n]=0) and štampa then
53 #                   writeln(' ');
54 #                 if (d[n]=0) and štampa then
55 #                   writeln(' ');
56 #                 if (d[n]=0) and štampa then
57 #                   writeln(' ');
58 #                 if (d[n]=0) and štampa then
59 #                   writeln(' ');
60 #                 if (d[n]=0) and štampa then
61 #                   writeln(' ');
62 #                 if (d[n]=0) and štampa then
63 #                   writeln(' ');
64 #                 if (d[n]=0) and štampa then
65 #                   writeln(' ');
66 #                 if (d[n]=0) and štampa then
67 #                   writeln(' ');
68 #                 if (d[n]=0) and štampa then
69 #                   writeln(' ');
70 #                 if (d[n]=0) and štampa then
71 #                   writeln(' ');
72 #                 if (d[n]=0) and štampa then
73 #                   writeln(' ');
74 #                 if (d[n]=0) and štampa then
75 #                   writeln(' ');
76 #                 if (d[n]=0) and štampa then
77 #                   writeln(' ');
78 #                 if (d[n]=0) and štampa then
79 #                   writeln(' ');
80 #                 if (d[n]=0) and štampa then
81 #                   writeln(' ');
82 #                 if (d[n]=0) and štampa then
83 #                   writeln(' ');
84 #                 if (d[n]=0) and štampa then
85 #                   writeln(' ');
86 #                 if (d[n]=0) and štampa then
87 #                   writeln(' ');
88 #                 if (d[n]=0) and štampa then
89 #                   writeln(' ');
90 #                 if (d[n]=0) and štampa then
91 #                   writeln(' ');
92 #                 if (d[n]=0) and štampa then
93 #                   writeln(' ');
94 #                 if (d[n]=0) and štampa then
95 #                   writeln(' ');
96 #                 if (d[n]=0) and štampa then
97 #                   writeln(' ');
98 #                 if (d[n]=0) and štampa then
99 #                   writeln(' ');
100 #                 if (d[n]=0) and štampa then
101 #                   writeln(' ');
102 #                 if (d[n]=0) and štampa then
103 #                   writeln(' ');
104 #                 if (d[n]=0) and štampa then
105 #                   writeln(' ');
106 #                 if (d[n]=0) and štampa then
107 #                   writeln(' ');
108 #                 if (d[n]=0) and štampa then
109 #                   writeln(' ');
110 #                 if (d[n]=0) and štampa then
111 #                   writeln(' ');
112 #                 if (d[n]=0) and štampa then
113 #                   writeln(' ');
114 #                 if (d[n]=0) and štampa then
115 #                   writeln(' ');
116 #                 if (d[n]=0) and štampa then
117 #                   writeln(' ');
118 #                 if (d[n]=0) and štampa then
119 #                   writeln(' ');
120 #                 if (d[n]=0) and štampa then
121 #                   writeln(' ');
122 #                 if (d[n]=0) and štampa then
123 #                   writeln(' ');
124 #                 if (d[n]=0) and štampa then
125 #                   writeln(' ');
126 #                 if (d[n]=0) and štampa then
127 #                   writeln(' ');
128 #                 if (d[n]=0) and štampa then
129 #                   writeln(' ');
130 #                 if (d[n]=0) and štampa then
131 #                   writeln(' ');
132 #                 if (d[n]=0) and štampa then
133 #                   writeln(' ');
134 #                 if (d[n]=0) and štampa then
135 #                   writeln(' ');
136 #                 if (d[n]=0) and štampa then
137 #                   writeln(' ');
138 #                 if (d[n]=0) and štampa then
139 #                   writeln(' ');
140 #                 if (d[n]=0) and štampa then
141 #                   writeln(' ');
142 #                 if (d[n]=0) and štampa then
143 #                   writeln(' ');
144 #                 if (d[n]=0) and štampa then
145 #                   writeln(' ');
146 #                 if (d[n]=0) and štampa then
147 #                   writeln(' ');
148 #                 if (d[n]=0) and štampa then
149 #                   writeln(' ');
150 #                 if (d[n]=0) and štampa then
151 #                   writeln(' ');
152 #                 if (d[n]=0) and štampa then
153 #                   writeln(' ');
154 #                 if (d[n]=0) and štampa then
155 #                   writeln(' ');
156 #                 if (d[n]=0) and štampa then
157 #                   writeln(' ');
158 #                 if (d[n]=0) and štampa then
159 #                   writeln(' ');
160 #                 if (d[n]=0) and štampa then
161 #                   writeln(' ');
162 #                 if (d[n]=0) and štampa then
163 #                   writeln(' ');
164 #                 if (d[n]=0) and štampa then
165 #                   writeln(' ');
166 #                 if (d[n]=0) and štampa then
167 #                   writeln(' ');
168 #                 if (d[n]=0) and štampa then
169 #                   writeln(' ');
170 #                 if (d[n]=0) and štampa then
171 #                   writeln(' ');
172 #                 if (d[n]=0) and štampa then
173 #                   writeln(' ');
174 #                 if (d[n]=0) and štampa then
175 #                   writeln(' ');
176 #                 if (d[n]=0) and štampa then
177 #                   writeln(' ');
178 #                 if (d[n]=0) and štampa then
179 #                   writeln(' ');
180 #                 if (d[n]=0) and štampa then
181 #                   writeln(' ');
182 #                 if (d[n]=0) and štampa then
183 #                   writeln(' ');
184 #                 if (d[n]=0) and štampa then
185 #                   writeln(' ');
186 #                 if (d[n]=0) and štampa then
187 #                   writeln(' ');
188 #                 if (d[n]=0) and štampa then
189 #                   writeln(' ');
190 #                 if (d[n]=0) and štampa then
191 #                   writeln(' ');
192 #                 if (d[n]=0) and štampa then
193 #                   writeln(' ');
194 #                 if (d[n]=0) and štampa then
195 #                   writeln(' ');
196 #                 if (d[n]=0) and štampa then
197 #                   writeln(' ');
198 #                 if (d[n]=0) and štampa then
199 #                   writeln(' ');
200 #                 if (d[n]=0) and štampa then
201 #                   writeln(' ');
202 #                 if (d[n]=0) and štampa then
203 #                   writeln(' ');
204 #                 if (d[n]=0) and štampa then
205 #                   writeln(' ');
206 #                 if (d[n]=0) and štampa then
207 #                   writeln(' ');
208 #                 if (d[n]=0) and štampa then
209 #                   writeln(' ');
210 #                 if (d[n]=0) and štampa then
211 #                   writeln(' ');
212 #                 if (d[n]=0) and štampa then
213 #                   writeln(' ');
214 #                 if (d[n]=0) and štampa then
215 #                   writeln(' ');
216 #                 if (d[n]=0) and štampa then
217 #                   writeln(' ');
218 #                 if (d[n]=0) and štampa then
219 #                   writeln(' ');
220 #                 if (d[n]=0) and štampa then
221 #                   writeln(' ');
222 #                 if (d[n]=0) and štampa then
223 #                   writeln(' ');
224 #                 if (d[n]=0) and štampa then
225 #                   writeln(' ');
226 #                 if (d[n]=0) and štampa then
227 #                   writeln(' ');
228 #                 if (d[n]=0) and štampa then
229 #                   writeln(' ');
230 #                 if (d[n]=0) and štampa then
231 #                   writeln(' ');
232 #                 if (d[n]=0) and štampa then
233 #                   writeln(' ');
234 #                 if (d[n]=0) and štampa then
235 #                   writeln(' ');
236 #                 if (d[n]=0) and štampa then
237 #                   writeln(' ');
238 #                 if (d[n]=0) and štampa then
239 #                   writeln(' ');
240 #                 if (d[n]=0) and štampa then
241 #                   writeln(' ');
242 #                 if (d[n]=0) and štampa then
243 #                   writeln(' ');
244 #                 if (d[n]=0) and štampa then
245 #                   writeln(' ');
246 #                 if (d[n]=0) and štampa then
247 #                   writeln(' ');
248 #                 if (d[n]=0) and štampa then
249 #                   writeln(' ');
250 #                 if (d[n]=0) and štampa then
251 #                   writeln(' ');
252 #                 if (d[n]=0) and štampa then
253 #                   writeln(' ');
254 #                 if (d[n]=0) and štampa then
255 #                   writeln(' ');
256 #                 if (d[n]=0) and štampa then
257 #                   writeln(' ');
258 #                 if (d[n]=0) and štampa then
259 #                   writeln(' ');
260 #                 if (d[n]=0) and štampa then
261 #                   writeln(' ');
262 #                 if (d[n]=0) and štampa then
263 #                   writeln(' ');
264 #                 if (d[n]=0) and štampa then
265 #                   writeln(' ');
266 #                 if (d[n]=0) and štampa then
267 #                   writeln(' ');
268 #                 if (d[n]=0) and štampa then
269 #                   writeln(' ');
270 #                 if (d[n]=0) and štampa then
271 #                   writeln(' ');
272 #                 if (d[n]=0) and štampa then
273 #                   writeln(' ');
274 #                 if (d[n]=0) and štampa then
275 #                   writeln(' ');
276 #                 if (d[n]=0) and štampa then
277 #                   writeln(' ');
278 #                 if (d[n]=0) and štampa then
279 #                   writeln(' ');
280 #                 if (d[n]=0) and štampa then
281 #                   writeln(' ');
282 #                 if (d[n]=0) and štampa then
283 #                   writeln(' ');
284 #                 if (d[n]=0) and štampa then
285 #                   writeln(' ');
286 #                 if (d[n]=0) and štampa then
287 #                   writeln(' ');
288 #                 if (d[n]=0) and štampa then
289 #                   writeln(' ');
290 #                 if (d[n]=0) and štampa then
291 #                   writeln(' ');
292 #                 if (d[n]=0) and štampa then
293 #                   writeln(' ');
294 #                 if (d[n]=0) and štampa then
295 #                   writeln(' ');
296 #                 if (d[n]=0) and štampa then
297 #                   writeln(' ');
298 #                 if (d[n]=0) and štampa then
299 #                   writeln(' ');
300 #                 if (d[n]=0) and štampa then
301 #                   writeln(' ');
302 #                 if (d[n]=0) and štampa then
303 #                   writeln(' ');
304 #                 if (d[n]=0) and štampa then
305 #                   writeln(' ');
306 #                 if (d[n]=0) and štampa then
307 #                   writeln(' ');
308 #                 if (d[n]=0) and štampa then
309 #                   writeln(' ');
310 #                 if (d[n]=0) and štampa then
311 #                   writeln(' ');
312 #                 if (d[n]=0) and štampa then
313 #                   writeln(' ');
314 #                 if (d[n]=0) and štampa then
315 #                   writeln(' ');
316 #                 if (d[n]=0) and štampa then
317 #                   writeln(' ');
318 #                 if (d[n]=0) and štampa then
319 #                   writeln(' ');
320 #                 if (d[n]=0) and štampa then
321 #                   writeln(' ');
322 #                 if (d[n]=0) and štampa then
323 #                   writeln(' ');
324 #                 if (d[n]=0) and štampa then
325 #                   writeln(' ');
326 #                 if (d[n]=0) and štampa then
327 #                   writeln(' ');
328 #                 if (d[n]=0) and štampa then
329 #                   writeln(' ');
330 #                 if (d[n]=0) and štampa then
331 #                   writeln(' ');
332 #                 if (d[n]=0) and štampa then
333 #                   writeln(' ');
334 #                 if (d[n]=0) and štampa then
335 #                   writeln(' ');
336 #                 if (d[n]=0) and štampa then
337 #                   writeln(' ');
338 #                 if (d[n]=0) and štampa then
339 #                   writeln(' ');
340 #                 if (d[n]=0) and štampa then
341 #                   writeln(' ');
342 #                 if (d[n]=0) and štampa then
343 #                   writeln(' ');
344 #                 if (d[n]=0) and štampa then
345 #                   writeln(' ');
346 #                 if (d[n]=0) and štampa then
347 #                   writeln(' ');
348 #                 if (d[n]=0) and štampa then
349 #                   writeln(' ');
350 #                 if (d[n]=0) and štampa then
351 #                   writeln(' ');
352 #                 if (d[n]=0) and štampa then
353 #                   writeln(' ');
354 #                 if (d[n]=0) and štampa then
355 #                   writeln(' ');
356 #                 if (d[n]=0) and štampa then
357 #                   writeln(' ');
358 #                 if (d[n]=0) and štampa then
359 #                   writeln(' ');
360 #                 if (d[n]=0) and štampa then
361 #                   writeln(' ');
362 #                 if (d[n]=0) and štampa then
363 #                   writeln(' ');
364 #                 if (d[n]=0) and štampa then
365 #                   writeln(' ');
366 #                 if (d[n]=0) and štampa then
367 #                   writeln(' ');
368 #                 if (d[n]=0) and štampa then
369 #                   writeln(' ');
370 #                 if (d[n]=0) and štampa then
371 #                   writeln(' ');
372 #                 if (d[n]=0) and štampa then
373 #                   writeln(' ');
374 #                 if (d[n]=0) and štampa then
375 #                   writeln(' ');
376 #                 if (d[n]=0) and štampa then
377 #                   writeln(' ');
378 #                 if (d[n]=0) and štampa then
379 #                   writeln(' ');
380 #                 if (d[n]=0) and štampa then
381 #                   writeln(' ');
382 #                 if (d[n]=0) and štampa then
383 #                   writeln(' ');
384 #                 if (d[n]=0) and štampa then
385 #                   writeln(' ');
386 #                 if (d[n]=0) and štampa then
387 #                   writeln(' ');
388 #                 if (d[n]=0) and štampa then
389 #                   writeln(' ');
390 #                 if (d[n]=0) and štampa then
391 #                   writeln(' ');
392 #                 if (d[n]=0) and štampa then
393 #                   writeln(' ');
394 #                 if (d[n]=0) and štampa then
395 #                   writeln(' ');
396 #                 if (d[n]=0) and štampa then
397 #                   writeln(' ');
398 #                 if (d[n]=0) and štampa then
399 #                   writeln(' ');
400 #                 if (d[n]=0) and štampa then
401 #                   writeln(' ');
402 #                 if (d[n]=0) and štampa then
403 #                   writeln(' ');
404 #                 if (d[n]=0) and štampa then
405 #                   writeln(' ');
406 #                 if (d[n]=0) and štampa then
407 #                   writeln(' ');
408 #                 if (d[n]=0) and štampa then
409 #                   writeln(' ');
410 #                 if (d[n]=0) and štampa then
411 #                   writeln(' ');
412 #                 if (d[n]=0) and štampa then
413 #                   writeln(' ');
414 #                 if (d[n]=0) and štampa then
415 #                   writeln(' ');
416 #                 if (d[n]=0) and štampa then
417 #                   writeln(' ');
418 #                 if (d[n]=0) and štampa then
419 #                   writeln(' ');
420 #                 if (d[n]=0) and štampa then
421 #                   writeln(' ');
422 #                 if (d[n]=0) and štampa then
423 #                   writeln(' ');
424 #                 if (d[n]=0) and štampa then
425 #                   writeln(' ');
426 #                 if (d[n]=0) and štampa then
427 #                   writeln(' ');
428 #                 if (d[n]=0) and štampa then
429 #                   writeln(' ');
430 #                 if (d[n]=0) and štampa then
431 #                   writeln(' ');
432 #                 if (d[n]=0) and štampa then
433 #                   writeln(' ');
434 #                 if (d[n]=0) and štampa then
435 #                   writeln(' ');
436 #                 if (d[n]=0) and štampa then
437 #                   writeln(' ');
438 #                 if (d[n]=0) and štampa then
439 #                   writeln(' ');
440 #                 if (d[n]=0) and štampa then
441 #                   writeln(' ');
442 #                 if (d[n]=0) and štampa then
443 #                   writeln(' ');
444 #                 if (d[n]=0) and štampa then
445 #                   writeln(' ');
446 #                 if (d[n]=0) and štampa then
447 #                   writeln(' ');
448 #                 if (d[n]=0) and štampa then
449 #                   writeln(' ');
450 #                 if (d[n]=0) and štampa then
451 #                   writeln(' ');
452 #                 if (d[n]=0) and štampa then
453 #                   writeln(' ');
454 #                 if (d[n]=0) and štampa then
455 #                   writeln(' ');
456 #                 if (d[n]=0) and štampa then
457 #                   writeln(' ');
458 #                 if (d[n]=0) and štampa then
459 #                   writeln(' ');
460 #                 if (d[n]=0) and štampa then
461 #                   writeln(' ');
462 #                 if (d[n]=0) and štampa then
463 #                   writeln(' ');
464 #                 if (d[n]=0) and štampa then
465 #                   writeln(' ');
466 #                 if (d[n]=0) and štampa then
467 #                   writeln(' ');
468 #                 if (d[n]=0) and štampa then
469 #                   writeln(' ');
470 #                 if (d[n]=0) and štampa then
471 #                   writeln(' ');
472 #                 if (d[n]=0) and štampa then
473 #                   writeln(' ');
474 #                 if (d[n]=0) and štampa then
475 #                   writeln(' ');
476 #                 if (d[n]=0) and štampa then
477 #                   writeln(' ');
478 #                 if (d[n]=0) and štampa then
479 #                   writeln(' ');
480 #                 if (d[n]=0) and štampa then
481 #                   writeln(' ');
482 #                 if (d[n]=0) and štampa then
483 #                   writeln(' ');
484 #                 if (d[n]=0) and štampa then
485 #                   writeln(' ');
486 #                 if (d[n]=0) and štampa then
487 #                   writeln(' ');
488 #                 if (d[n]=0) and štampa then
489 #                   writeln(' ');
490 #                 if (d[n]=0) and štampa then
491 #                   writeln(' ');
492 #                 if (d[n]=0) and štampa then
493 #                   writeln(' ');
494 #                 if (d[n]=0) and štampa then
495 #                   writeln(' ');
496 #                 if (d[n]=0) and štampa then
497 #                   writeln(' ');
498 #                 if (d[n]=0) and štampa then
499 #                   writeln(' ');
500 #                 if (d[n]=0) and štampa then
501 #                   writeln(' ');
502 #                 if (d[n]=0) and štampa then
503 #                   writeln(' ');
504 #                 if (d[n]=0) and štampa then
505 #                   writeln(' ');
506 #                 if (d[n]=0) and štampa then
507 #                   writeln(' ');
508 #                 if (d[n]=0) and štampa then
509 #                   writeln(' ');
510 #                 if (d[n]=0) and štampa then
511 #                   writeln(' ');
512 #                 if (d[n]=0) and štampa then
513 #                   writeln(' ');
514 #                 if (d[n]=0) and štampa then
515 #                   writeln(' ');
516 #                 if (d[n]=0) and štampa then
517 #                   writeln(' ');
518 #                 if (d[n]=0) and štampa then
519 #                   writeln(' ');
520 #                 if (d[n]=0) and štampa then
521 #                   writeln(' ');
522 #                 if (d[n]=0) and štampa then
523 #                   writeln(' ');
524 #                 if (d[n]=0) and štampa then
525 #                   writeln(' ');
526 #                 if (d[n]=0) and štampa then
527 #                   writeln(' ');
528 #                 if (d[n]=0) and štampa then
529 #                   writeln(' ');
530 #                 if (d[n]=0) and štampa then
531 #                   writeln(' ');
532 #                 if (d[n]=0) and štampa then
533 #                   writeln(' ');
534 #                 if (d[n]=0) and štampa then
535 #                   writeln(' ');
536 #                 if (d[n]=0) and štampa then
537 #                   writeln(' ');
538 #                 if (d[n]=0) and štampa then
539 #                   writeln(' ');
540 #                 if (d[n]=0) and štampa then
541 #                   writeln(' ');
542 #                 if (d[n]=0) and štampa then
543 #                   writeln(' ');
544 #                 if (d[n]=0) and štampa then
545 #                   writeln(' ');
546 #                 if (d[n]=0) and štampa then
547 #                   writeln(' ');
548 #                 if (d[n]=0) and štampa then
549 #                   writeln(' ');
550 #                 if (d[n]=0) and štampa then
551 #                   writeln(' ');
552 #                 if (d[n]=0) and štampa then
553 #                   writeln(' ');
554 #                 if (d[n]=0) and štampa then
555 #                   writeln(' ');
556 #                 if (d[n]=0) and štampa then
557 #                   writeln(' ');
558 #                 if (d[n]=0) and štampa then
559 #                   writeln(' ');
560 #                 if (d[n]=0) and štampa then
561 #                   writeln(' ');
562 #                 if (d[n]=0) and štampa then
563 #                   writeln(' ');
564 #                 if (d[n]=0) and štampa then
565 #                   writeln(' ');
566 #                 if (d[n]=0) and štampa then
567 #                   writeln(' ');
568 #                 if (d[n]=0) and štampa then
569 #                   writeln(' ');
570 #                 if (d[n]=0) and štampa then
571 #                   writeln(' ');
572 #                 if (d[n]=0) and štampa then
573 #                   writeln(' ');
574 #                 if (d[n]=0) and štampa then
575 #                   writeln(' ');
576 #                 if (d[n]=0) and štampa then
577 #                   writeln(' ');
578 #                 if (d[n]=0) and štampa then
579 #                   writeln(' ');
580 #                 if (d[n]=0) and štampa then
581 #                   writeln(' ');
582 #                 if (d[n]=0) and štampa then
583 #                   writeln(' ');
584 #                 if (d[n]=0) and štampa then
585 #                   writeln(' ');
586 #                 if (d[n]=0) and štampa then
587 #                   writeln(' ');
588 #                 if (d[n]=0) and štampa then
589 #                   writeln(' ');
590 #                 if (d[n]=0) and štampa then
591 #                   writeln(' ');
592 #                 if (d[n]=0) and štampa then
593 #                   writeln(' ');
594 #                 if (d[n]=0) and štampa then
595 #                   writeln(' ');
596 #                 if (d[n]=0) and štampa then
597 #                   writeln(' ');
598 #                 if (d[n]=0) and štampa then
599 #                   writeln(' ');
600 #                 if (d[n]=0) and štampa then
601 #                   writeln(' ');
602 #                 if (d[n]=0) and štampa then
603 #                   writeln(' ');
604 #                 if (d[n]=0) and štampa then
605 #                   writeln(' ');
606 #                 if (d[n]=0) and štampa then
607 #                   writeln(' ');
608 #                 if (d[n]=0) and štampa then
609 #                   writeln(' ');
610 #                 if (d[n]=0) and štampa then
611 #                   writeln(' ');
612 #                 if (d[n]=0) and štampa then
613 #                   writeln(' ');
614 #                 if (d[n]=0) and štampa then
615 #                   writeln(' ');
616 #                 if (d[n]=0) and štampa then
617 #                   writeln(' ');
618 #                 if (d[n]=0) and štampa then
619 #                   writeln(' ');
620 #                 if (d[n]=0) and štampa then
621 #                   writeln(' ');
622 #                 if (d[n]=0) and štampa then
623 #                   writeln(' ');
624 #                 if (d[n]=0) and štampa then
625 #                   writeln(' ');
626 #                 if (d[n]=0) and štampa then
627 #                   writeln(' ');
628 #                 if (d[n]=0) and štampa then
629 #                   writeln(' ');
630 #                 if (d[n]=0) and štampa then
631 #                   writeln(' ');
632 #                 if (d[n]=0) and štampa then
633 #                   writeln(' ');
634 #                 if (d[n]=0) and štampa then
635 #                   writeln(' ');
636 #                 if (d[n]=0) and štampa then
637 #                   writeln(' ');
638 #                 if (d[n]=0) and štampa then
639 #                   writeln(' ');
640 #                 if (d[n]=0) and štampa then
641 #                   writeln(' ');
642 #                 if (d[n]=0) and štampa then
643 #                   writeln(' ');
644 #                 if (d[n]=0) and štampa then
645 #                   writeln(' ');
646 #                 if (d[n]=0) and štampa then
647 #                   writeln(' ');
648 #                 if (d[n]=0) and štampa then
649 #                   writeln(' ');
650 #                 if (d[n]=0) and štampa then
651 #                   writeln(' ');
652 #                 if (d[n]=0) and štampa then
653 #                   writeln(' ');
654 #                 if (d[n]=0) and štampa then
655 #                   writeln(' ');
656 #                 if (d[n]=0) and štampa then
657 #                   writeln(' ');
658 #                 if (d[n]=0) and štampa then
659 #                   writeln(' ');
660 #                 if (d[n]=0) and štampa then
661 #                   writeln(' ');
662 #                 if (d[n]=0) and štampa then
663 #                   writeln(' ');
664 #                 if (d[n]=0) and štampa then
665 #                   writeln(' ');
666 #                 if (d[n]=0) and štampa then
667 #                   writeln(' ');
668 #                 if (d[n]=0) and štampa then
669 #                   writeln(' ');
670 #                 if (d[n]=0) and štampa then
671 #                   writeln(' ');
672 #                 if (d[n]=0) and štampa then
673 #                   writeln(' ');
674 #                 if (d[n]=0) and štampa then
675 #                   writeln(' ');
676 #                 if (d[n]=0) and štampa then
677 #                   writeln(' ');
678 #                 if (d[n]=0) and štampa then
679 #                   writeln(' ');
680 #                 if (d[n]=0) and štampa then
681 #                   writeln(' ');
682 #                 if (d[n]=0) and štampa then
683 #                   writeln(' ');
684 #                 if (d[n]=0) and štampa then
685 #                   writeln(' ');
686 #                 if (d[n]=0) and štampa then
687 #                   writeln(' ');
688 #                 if (d[n]=0) and štampa then
689 #                   writeln(' ');
690 #                 if (d[n]=0) and štampa then
691 #                   writeln(' ');
692 #                 if (d[n]=0) and štampa then
693 #                   writeln(' ');
694 #                 if (d[n]=0) and štampa then
695 #                   writeln(' ');
696 #                 if (d[n]=0) and štampa then
697 #                   writeln(' ');
698 #                 if (d[n]=0) and štampa then
699 #                   writeln(' ');
700 #                 if (d[n]=0) and štampa then
701 #                   writeln(' ');
702 #                 if (d[n]=0) and štampa then
703 #                   writeln(' ');
704 #                 if (d[n]=0) and štampa then
705 #                   writeln(' ');
706 #                 if (d[n]=0) and štampa then
707 #                   writeln(' ');
708 #                 if (d[n]=0) and štampa then
709 #                   writeln(' ');
710 #                 if (d[n]=0) and štampa then
711 #                   writeln(' ');
712 #                 if (d[n]=0) and štampa then
713 #                   writeln(' ');
714 #                 if (d[n]=0) and štampa then
715 #                   writeln(' ');
716 #                 if (d[n]=0) and štampa then
717 #                   writeln(' ');
718 #                 if (d[n]=0) and štampa then
719 #                   writeln(' ');
720 #                 if (d[n]=0) and štampa then
721 #                   writeln(' ');
722 #                 if (d[n]=0) and štampa then
723 #                   writeln(' ');
724 #                 if (d[n]=0) and štampa then
725 #                   writeln(' ');
726 #                 if (d[n]=0) and štampa then
727 #                   writeln(' ');
728 #                 if (d[n]=0) and štampa then
729 #                   writeln(' ');
730 #                 if (d[n]=0) and štampa then
731 #                   writeln(' ');
732 #                 if (d[n]=0) and štampa then
733 #                   writeln(' ');
734 #                 if (d[n]=0) and štampa then
735 #                   writeln(' ');
736 #                 if (d[n]=0) and štampa then
737 #                   writeln(' ');
738 #                 if (d[n]=0) and štampa then
739 #                   writeln(' ');
740 #                 if (d[n]=0) and štampa then
741 #                   writeln(' ');
742 #                 if (d[n]=0) and štampa then
743 #                   writeln(' ');
744 #                 if (d[n]=0) and štampa then
745 #                   writeln(' ');
746 #                 if (d[n]=0) and štampa then
747 #                   writeln(' ');
748 #                 if (d[n]=0) and štampa then
749 #                   writeln(' ');
750 #                 if (d[n]=0) and štampa then
751 #                   writeln(' ');
752 #                 if (d[n]=0) and štampa then
753 #                   writeln(' ');
754 #                 if (d[n]=0) and štampa then
755 #                   writeln(' ');
756 #                 if (d[n]=0) and štampa then
757 #                   writeln(' ');
758 #                 if (d[n]=0) and štampa then
759 #                   writeln(' ');
760 #                 if (d[n]=0) and štampa then
761 #                   writeln(' ');
762 #                 if (d[n]=0) and štampa then
763 #                   writeln(' ');
764 #                 if (d[n]=0) and štampa then
765 #                   writeln(' ');
766 #                 if (d[n]=0) and štampa then
767 #                   writeln(' ');
768 #                 if (d[n]=0) and štampa then
769 #                   writeln(' ');
770 #                 if (d[n]=0) and štampa then
771 #                   writeln(' ');
772 #                 if (d[n]=0) and štampa then
773 #                   writeln(' ');
774 #                 if (d[n]=0) and štampa then
775 #                   writeln(' ');
776 #                 if (d[n]=0) and štampa then
777 #                   writeln(' ');
778 #                 if (d[n]=0) and štampa then
779 #                   writeln(' ');
780 #                 if (d[n]=0) and štampa then
781 #                   writeln(' ');
782 #                 if (d[n]=0) and štampa then
783 #                   writeln(' ');
784 #                 if (d[n]=0) and štampa then
785 #                   writeln(' ');
786 #                 if (d[n]=0) and štampa then
787 #                   writeln(' ');
788 #                 if (d[n]=0) and štampa then
789 #                   writeln(' ');
790 #                 if (d[n]=0) and štampa then
791 #                   writeln(' ');
792 #                 if (d[n]=0) and štampa then
793 #                   writeln(' ');
794 #                 if (d[n]=0) and štampa then
795 #                   writeln(' ');
796 #                 if (d[n]=0) and štampa then
797 #                   writeln(' ');
798 #                 if (d[n]=0) and štampa then
799 #                   writeln(' ');
800 #                 if (d[n]=0) and štampa then
801 #                   writeln(' ');
802 #                 if (d[n]=0) and štampa then
803 #                   writeln(' ');
804 #                 if (d[n]=0) and štampa then
805 #                   writeln(' ');
806 #                 if (d[n]=0) and štampa then
807 #                   writeln(' ');
808 #                 if (d[n]=0) and štampa then
809 #                   writeln(' ');
810 #                 if (d[n]=0) and štampa then
811 #                   writeln(' ');
812 #                 if (d[n]=0) and štampa then
813 #                   writeln(' ');
814 #                 if (d[n]=0) and štampa then
815 #                   writeln(' ');
816 #                 if (d[n]=0) and štampa then
817 #                   writeln(' ');
818 #                 if (d[n]=0) and štampa then
819 #                   writeln(' ');
820 #                 if (d[n]=0) and štampa then
821 #                   writeln(' ');
822 #                 if (d[n]=0) and štampa then
823 #                   writeln(' ');
824 #                 if (d[n]=0) and štampa then
825 #                   writeln(' ');
826 #                 if (d[n]=0) and štampa then
827 #                   writeln(' ');
828 #                 if (d[n]=0) and štampa then
829 #                   writeln(' ');
830 #                 if (d[n]=0) and štampa then
831 #                   writeln(' ');
832 #                 if (d[n]=0) and štampa then
833 #                   writeln(' ');
834 #                 if (d[n]=0) and štampa then
835 #                   writeln(' ');
836 #                 if (d[n]=0) and štampa then
837 #                   writeln(' ');
838 #                 if (d[n]=0) and štampa then
839 #                   writeln(' ');
840 #                 if (d[n]=0) and štampa then
841 #                   writeln(' ');
842 #                 if (d[n]=0) and štampa then
843 #                   writeln(' ');
844 #                 if (d[n]=0) and štampa then
845 #                   writeln(' ');
846 #                 if (d[n]=
```

PROSTE KOMBINACIJE DUŽINE 8

2	2	2	2	2	2	7	7	7	7	7	7
23	23	29	29	29	29	87	87	87	87	87	87
523	823	829	829	829	829	387	387	687	687	879	879
7523	8231	6829	7829	7829	7829	2387	2387	2487	4793	4793	9823
87523	58231	16829	67829	67829	67829	12387	92387	12687	24793	24793	59823
987523	582317	516829	567829	567829	567829	123879	923871	912687	124793	124793	159823
4987523	5823179	4516829	1567829	1567829	1567829	5123879	5123879	5923871	3912687	6124793	2159823
64987523	65823179	45168293	15678293	15678293	15678293	65123879	65923871	53912687	56124793	86124793	82159823
3	3	3	3	3	3	7	7	7	7	7	7
13	23	23	23	23	23	87	87	87	87	87	87
613	523	823	283	283	283	987	987	867	867	967	179
9613	7523	8231	1283	2637	9283	9873	9873	2467	2467	4967	2179
96137	87523	58231	61283	52837	79283	59473	59473	24671	24671	84667	42179
296137	987523	582317	961283	652837	579283	159473	159473	824671	824671	184667	342179
4296137	4987523	5823179	5961283	162837	879283	6159473	6159473	9824671	9824671	2184667	6342179
84296137	64987523	65823179	45961283	9162837	64579283	26159473	26159473	98246713	39824671	32184667	95342179
8	7	7	7	7	7	7	7	7	7	7	7
3	71	71	71	17	17	79	79	79	79	79	79
283	571	719	719	179	179	179	179	479	479	479	479
9283	6571	6571	7193	2179	5179	45179	45179	8793	8793	8793	8793
79283	36571	65719	57193	42179	45179	45179	45179	24793	24793	24793	24793
579283	426571	657193	457193	342179	451793	451793	451793	124793	124793	124793	124793
46579283	92436571	4657193	4657193	6342179	281793	2451793	6124793	6124793	6124793	6124793	6124793

slika 2

! Početni broj	!	2	!	3	!	5	!	7	!
!	!	!	!	!	!	!	!	!	!
! Dužina 3	!	2	!	3	!	1	!	1	!
! Dužina 4	!	4	!	6	!	2	!	16	!
! Dužina 5	!	7	!	43	!	13	!	36	!
! Dužina 6	!	13	!	35	!	5	!	73	!
! Dužina 7	!	7	!	24	!	3	!	42	!
! Dužina 8	!	6	!	7	!	-	!	20	!

slika 3

uvek — 140. liniju je trebalo dopuniti znamenom vrednosti D i E, a našu uslužnu disketu programom za generisanje permutacija!

**Permutacije sa ponavljanjem** su sasvim slične, a ipak malo drugačije: među elementima koji se raspoređuju u i jednakih. Pokušajmo, na primer, da rasporedimo A, B i B. Ako generišemo „obične“ permutacije, dobijemo AAB, ABA, AAB, ABA, BAA, BAA — šest rasporeda od kojih su tri puta po dva jednaka. Ukoliko, uopšte, imamo N elemenata, od toga N1 imaju jednu vrednost, N2 drugu i tako dalje, permutacija sa ponavljanjem će biti  $N! / (N1! N2! \dots)$ ; proverite li u različitim rasporeda elemenata A, A, A, B, C i C ima  $6! / (3! 2!) = 60$ .

**Varijacije sa ponavljanjem** dobro poznaju svi igrači sportske prognoze. Imamo, na primer, tri elementa, ti, tipove 0, 1 i 2 (N=3) od kojih treba formirati trinaestocifrene nizove (M=13) — kažemo da formiramo varijacije sa ponavljanjem trinaeste klase od tri elementa. Takvih varijacija ima  $3^{13} = 1.594.323$ . Sami izračunajte koliko bi ovakav sistem (koji garantuje tačno jednu trinaesticifru) koštao; ni čemu u međuvremenu ispisati sve varijacije treće klase (M=3) od elemenata A i B (N=2). Varijacija bi trebalo da ima  $NM = 2^3 = 8$  što se pokazuje tačnim: AAA, AAB, ABA, BAA, ABB, BAB, BBA, BBB.

**Varijacije bez ponavljanja** se sasvim malo razlikuju od opisane sportske prognoze: u rezultujućem nizu ne smeju da se ponavljaju isti elementi što znači da bi od A, B, C i D (N=4) generisali svega 12 varijacija sa ponavljanjem druge klase

(M=2): AB, BA, AC, CA, AD, DA, BC, CB, BD, DB, CD i DC; da smo generisali varijacije sa ponavljanjem, bilo bi ih 42=16; trebalo bi dodati i AA, BB, CC i DD. Varijacija bez ponavljanja od N elemenata M-te klase ima ukupno  $N(N-1)(N-2) \dots (N-M+1)$ .

**Kombinacije bez ponavljanja** su popularne među igračima LOTO-a baš kao što su varijacije sa ponavljanjem osnova sportske prognoze: generišemo varijacije sa ponavljanjem kod kojih redosled elemenata nije bitan — smatramo da su sekvence AB i BA identične! Kombinacija sa ponavljanjem od N elemenata M-te klase ima tačno  $N! / (M! (N-M)!)$ . Pokušajmo, na primer, da generišemo varijacije bez ponavljanja treće klase (M=3) od pet elemenata (N=5) A, B, C, D, E: trebalo bi da ih bude  $5! / (3! 2!) = 10$ . Zaista: ABC, ABD, ABE, ACE, ADE, BCD, BCE, BDE, CDE. Sami izračunajte koliko bi koštao pun LOTO sistem koji bi garantovao da će se od 39 brojeva (N=39) izvući baš vaših 7 (M=7); takav bi sistem imao oko 15.381.000 kombinacija.

**Kombinacije sa ponavljanjem** nisu predmet ovog zadatka jer nismo uspeali da smislimo neku njihovu veće praktičnu primenu; verujemo, ipak, da ćete lako zaključiti o čemu se radi.

Slika 1 predstavlja malo obezbeđenje od štamparskih grešaka: na njoj vidimo koliko se permutacija, varijacija i kombinacija može generisati od zadatah N elemenata.

**Razmatranje problema . . .**

Već ste svakako pogodili da će pisanje programa koji generišu razne do sada pomenute nizove biti vaš zadatak. Sa koje strane da „napadnete problem“? Najlakše

rešenje je generisati varijacije sa ponavljanjem kao na slici 2: varijacije K-te klase od N elemenata su zapravo svi K-cifreni brojevi u sistemu koji se sastoji od N cifara. Pošto umete da generišete sve varijacije sa ponavljanjem, iz njih možete da izdvojite sve ostale strukture: generišete varijaciju po varijaciju, odbacujete one u kojima se neki elementi ponavljaju i tako dobijate varijacije bez ponavljanja ili, za N=K, permutacije; ni generisanje kombinacija nije mnogo komplikovanije. Ovakvo bi rešenje, međutim, bilo krajnje loše — generišemo i pamtimo gomilu brojeva, a onda dobar deo njih odbacujemo. Predlažemo vam, dakle, da sami „izmislite“ algoritam koji generiše baš onu sekvencu koja vas interesuje i da pokušate da ga učinite što brzim. Teško je, naravno, očekivati da će samo jedan saradnik „Radionice“ izmisliti najbolje algoritme za sve sekvence, pa će naše konačno rešenje verovatno biti mešavina više programa raznih autora.

Iako svi programi ispisuju rezultate na ekranu, treba predvideti da će neko poželeti da ih koristi i kao potprograme, što znači da PRINT naredbe ne smeju da budu razbacane duž čitavog programa.

**Ništa bez listinga**

Kako treba da izgledaju prilozii za ovu rubriku? Komentarisani listing je najvažniji: na osnovu njega ćemo odabrati nekoliko programa koje ćemo detaljno testirati. Lepo je ako pošaljete i kasetu (disketu), ali ona ne predstavlja zamenu za listing!

Što se ostale dokumentacije tiče, znamo da njeno pripremanje predstavlja pravu muku pa nećemo preterivati u zahtevima — dovoljno je nekoliko reči o korišćenim algoritimima, po neka blok šema i, važnije od svega toga, vremena koja vaš program troši rešavajući probleme sa slike 3.

Sa zahvalnošću ćemo primiti i predloge za sledeće zadatke — ova rubrika, više od bilo koje druge, zavisi od vaše saradnje i vašeg interesovanja, što znači da je vaša saradnja jedini način da rubrika opstane i da se razvija. Šaljite zato vaše priloge na adresu „Računari“ (za „programersku radionicu“), Bulevar vojvode Mišića 17, Beograd tako da pristignu pre 1. aprila 1987.





## Mali oglasi

# Nove cene, nova pravila

Ako ne možete da podnesete da drugi nemaju ono što vi imate, objavite svoj mali oglas u „Računarima“.

Ako ne možete da podnesete da drugi imaju ono što vi nemate, javite se na neki od malih oglasa u „Računarima“.

Prva stvar koju treba da uradite jeste da se odlučite da li želite običan ili uokviren mali oglas.

### Obični oglasi

Cena običnog malog oglasa do dvadeset reči je 1800 dinara. Svaka naredna reč košta još 150 dinara. Veznici, predlozi, prilozii, zamenice, brojevi i ostale „male“ reči se računaju u cenu. Adresa oglašivača se ne računa u cenu. Tri važna ograničenja: mali oglas ne može biti duži od 50 reči, mali oglas ne može biti štampan velikim slovima i uz mali oglas se ne mogu objavljivati crteži i fotografije. Mali oglas koji se ne uklapa u ove okvire pripadaju kategoriji uokvirenih malih oglasa.

### Uokvireni mali oglasi

Cena uokvirenog malog oglasa je 2000 dinara i po visinskom centimetru u stupcu širine 9,5 cecera ako oglas nije viši od pet centimetara i 3000 dinara po visinskom centimetru ako je mali oglas visok između pet i deset centimetara. U sklopu uokvirenog malog oglasa mogu se objavljivati fotografije i crteži i mogu se birati veličina i tipovi slova (belo, polucрно, kurziv). Fotografije i crteži se plaćaju prema prostoru kao da se radi o tekstu. Jedno važno ograničenje: uokvireni oglasi preko 10 centimetara ne spadaju u kategoriju malih oglasa. Za njih važe pune komercijalne cene i oni se ugovaraju sa oglasnim odeljenjem BIGZ-a.

### Priprema malih oglasa

Poželjno je da mali oglas počinje sa Prodajem, Kupujem, Džrim časove, Menjam... i sličnim što ukazuje na njegovu sadržinu. Adresa oglašivača se kuca u produžetku teksta malog oglasa, a ne odvojeno. Ova ograničenja, razume se, ne važe za uokvirene male oglase.

Uz mali oglas treba navesti njegovu vrstu (običan, uokviren) i kategoriju u kojoj će biti objavljen „spektrum“, „komodor“, „hardver“, „literatura“...)

### Prijem i plaćanje malih oglasa

Mali oglas treba dostaviti na adresu redakcije „Računari“ — BIGZ (za male oglase), Bulevar vojvode Mišića 17, 11000 Beograd najkasnije do petog u mesecu. Sv oglas koji do ovog roka pristigao u redakciju poštom, lično i, uz određena ograničenja, telefonom, biće uvršćen u sledeći broj.

Mali oglas se, po pravilu, plaćaju unapred bankovnom uplatnicom na račun 60802-603-23264 BIGZ, Bulevar vojvode Mišića 17, 11000 Beograd, sa obezbeđenim naznakom: „Računari“, mali oglas. Kopiju uplatnice treba, obavezno, dostaviti zajedno sa tekstom malog oglasa.

### Prijem malih oglasa telefonom

Mali oglas se, u opravdanim slučajevima, može redakciji dostaviti i telefonom (011/850-161), ali samo pod uslovom da nije duži od dvadeset reči ili da se naručuje ponavljanje oglasa iz prethodnog broja „Računara“. U tom slučaju oglas se može platiti i naknadno, uplatnicom koju će oglašivaču uputiti redakcija.

### Važno upozorenje

Sadržaj malog oglasa predstavlja diskreciono pravo oglašivača, pod uslovom da ne dolazi u koliziju sa zakonom i da ne vredi javni moral. U tom smislu, u „Računarima“ se ne može objaviti mali oglas u kome se naglašava ili veliča pornografska, militaristička ili slična društvenoneprihvatljiva osobina pojedinih igara.

## SPEKTRUM

„SPEKTRUMOVCI“!!! Superkompleti najnovijih programa po staroj ceni od samo 700 din.+kasete. Superkomplet 38: DRUID, SCOOBY DOO, GREAT ESCAPE, COBRA, DANDY, EKRAANSKI EDITOR, GOONIES, STREET HAWK 2, FIRELORD, FLUIDRIDER, FAIRLIGHT 2 (2 programa)... Superkomplet 37: BOMBSQUARE, LIGHTFORGE, WAR 1.2, DEACTIVATORS... PRODIGY, BREAKTHRU, OLLI & LISSA, S. FORCE HARDIER, S. FORCE COBRA, CONQUEST, SKITTLES, ANTENE... Svi do sada izašli kompleti (40) za samo 12000 din. Garantujemo kvalitet!!!  
Peric Nenad, Braće Miladinov 12, 37000 KRUŠEVAC, Tel. (037) 33-510\* 031001

„SPEKTRUMOVCI!!! Superkompleti najnovijih programa po staroj ceni od samo 700 din.+ kasete. Superkomplet 40: MATCH DAY 2, WAY OF TIGER 2, GAUNTLET (4 programa), XEVIUS, ARCHAEOLOGIST, SPEED KING 2, THRUST, MOONCRESTA 2, ICE TEMPLE, SPACE HARRIER, TELEFONSKI IMENIK... Superkomplet 39: YIE AB KUNG FU 2, TARZAN, GALVAN, HARD-BALL, FAT WOM, ROOM 10, ASTROHORSKOP, CRYSTAL CASTLES, ROUGE TROOPER, KNOCK OUT, MOONLIGHT MADNESS, CUSTARD KID, BUMPSNAP SPIKE, TURBOOOPY...  
Peric Nenad, Braće Miladinov 12, 37000 KRUŠEVAC, Tel. (037) 33-510\* 031002

VRHUNSKO GRADEVINSKI PROGRAMI za Spektrum: OKVIRI, ROSTILJI, REŠETKE, DIMENZIONIRANJE, TEMELJI, ISKAZ ARMATURE i drugi. Za radne organizacije i pojedince. Besplatna katalog. Kozala 17, 51000 Rijek, tel. 051/517-291. 031011

SPEKTRUMOVCI!!! Najnoviji kompleti! Komplet sa kasetom 1200 dinara. Besplatna katalog. Komplet 3: DRUID, URIDIM, Asterix... Komplet 5: Green Beret, Bomb Jack, Samantha Fox... Komplet 6: Blade Runner, Movie, West bank...  
Potebica Zvonko, Žrnjka Jošilja br. 5/14, 71000 Sarajevo, Tel: (071) 646-093 031003

Legend soft Vam nudu završenu priliku da nabavite svetske hitove za Spektrum, po super niskim cenama (match day 2, shao lin, fuckoff, top gun... Svi noviji hitovi u besplatnom katalogu, pokloni, popusti isporuka za 24 časa. Dejan Stanković, Nas. Sretna Dučica ga. 1/15, 14000 Valjevo, tel. 014/36-540 031004

NIŠ-SOFTI Najnoviji, Najjefitniji, NABUŽI, NAIKVALITETNIJI programi za Vaš SPEKTRUM u gradul Komplet — 800 din! Pojedinačno — 100 din! Nišijama istog datul Snimci na JVC dekul Besplatna spiskal!  
PETROVIĆ CEDIMIR, Stanka Pounovića 48/7, Niš, tel: 018-23-802. 031012

Spektrumovci — OMEENSOFT vam nudu: Turbo 1, četiri programa u čiju četu pomoć igre učitači duplo brže. Cena programa sa uputstvom — 600 dinara. Turbo 2, kao turbo 1, programe učitava trostruko većom brzinom. Cena programa sa uputstvom — 800 dinara. Ako naručite oba programa cene je 1000 dinara.  
Petković, Dimitrija Tucovića 2/56, 11420 Smederevska Palanka (026) 36-818 031005

SEX... SEKI... SEKI...  
Sex komplet samo za odrasle. Mouse, strip poker, new strip pocer, sex mission, sex crime, strip game, violens sex, red light room, esti it, dirty movie, zodiak strip, olisa lisa, solo sex... Svi programi koji će Vas duže vreme držati uz ekran, zato napred — naručite ih. Cena—800 din.  
Dejan Stanković, Nas. Sretna Dučica gal. 1/15 Valjevo 14000, tel. 36-540 (014) 031013

SPEKTRUMOVCI NAJNOVIJE IGRE (KOMPLET 15: SCOOBY DOO, GREAT ESCAPE... KOMPLET 16: SPEED KING 2, ICE TEMPLE...) KOMPLET 750, A POJEDINAČNO 150 DINARA. USLUŽNI (LASER BASKET, ART STUDIO...) KOMPLET 900 DINARA. BESPLATAN KATALOG.  
VUJIĆ PREDRAG, BRATČE VUJIĆIĆA 5, 71000 SARAJEVO, TEL. (071) 524-069 031006

MEDO SOFTI Najnovije igre u kompletno 10 programa: Paperboy, Dan Dare, Ghost's Goblines, XARQ, Superman... +kasete=1200 d. Do izlaska još 2 kompleta.  
Medvešek Denis Gomiličkih žrtava bit 58213 Kaštel Gomilica 031007

QBS — programi za SPEKTRUM, od najstarijih do suvremenih, od sada pojedinačno (80 d.) i u kompletima (12 p. — 700 d.). Snimamo direktno iz spektruma, normalno ili TURBO TAPEOM. Za katalog poslati 100 d. QBS, Takovska 46, 32300 G. Milano-vas, Tel. 032/714-220. 031014

Prodajem A/D konverter za Spektrum (pogodan za softversko otklovlj i za uređaj za prepoznavanje govora) i kompjuterski kasetofon sa brojačem. Sve to menjam za mikrodizaj uz doplatu. Telefon 019-74-769\* 031008

Psychosoft — software for ZX Spektrum. Prvi donosimo na YU tržište Engleske hitove: Top Gun (Ocean), Super Soccer (imagine), Aliens (Alien 2) Snao Lin's Road (Korami)... i mnoge druge koji su već nekoliko godina na našoj top listi. Brzu i kvalitetnu uslugu Vam garantuje vrhunska oprema. Katalog je besplatan.  
Golubović Boban, Karadordeva 46/59, 11300 Smederevo, tel. 026/26-184 031015

PRODAJEMO najnovije i isključivo najbolje programe za spektrum. GAUNTLET, TOP GUN, SOCCER itd. Javite se za spisak. Kvalitet vrhunski.  
MILOŠ JELESJIJEVIĆ, ROZE LUKSEMBURGA 2/A, 11000 Beograd, TEL. 011/595-447 031009

SPEKTRUM 48k (nov) 2mj.+original kasetofon (nov)+350 programa na 30 kaseti +kvalitetna literatura=14 cm. Fatović Amel, Gavrića Principa 15, 79220 Bos. Novi, tel. 079/81-181 031010

SPEKTRUM SOFTWARE STUDIO  
Veliki izbor namenskih programa (programski jezici, assembleri, Disasembleri, kompajleri, mašinske rutine, bazi proširenja, grafički programi tekst procesori, baze podataka itd) — svaki sa uputstvom, kao i knjiga i priručnika za vaš SPEKTRUM. Sve sa uslovima prodaje možete poručiti na adresu:  
Patrić Mirko, Strahinjčića Bana 56, 11000 Beograd, tel. 011/188-190 i pošte 15h 031025







**SPECTRUMOVCI!!!** Jedini koji još uvijek uz najnižju cijenu programa (60-16kb i 80 din-48kb), snima iz Spectrumsa. Garantirano ispravak ispravan i nakon nekolicina godina. Uverite se! Uzi to stari, novi i najnoviji programi, popusti i besplatni katalog na adresi: Milivoje Branišić, Katskova 43, 54400 Opatjevac, Tel: (061/58-784) 031052

**ZAMIR SOFT:** Spectrumovci! Već treću godinu Zamir soft je za vama. Mnogi članovi kluba su se uvjerili u naše pogodnosti, uverite se i vi. Postoje mnogi razlozi da postanete član ZAMIR'S CLUBA, a to su profesionalne usluge vrhunskog kvaliteta brza isporuka, nikada cijene, stalni kontakti, novi programi i drugo. Kod nas možete dobiti programe pojedinačno ili u kompletima na kviljetnim kasetama. Član kluba postaje se neodrživo veom od 3000 din, a popusti za članove kluba je 30%. Zamir soft vam garantira kvalitetno i svakodnevno snimljenog programa. Katalog sa naznakom "SPECTRUM" tražite na adresi: Danijel Kurtović, Maršala Tita 72, 88000 Mostar ili na tel. (088) 53-644 031049



**SPECTRUMOVCI!!!**  
Za one koji se više ne igraju već ozbiljno koriste svoj ljubimca — **NAMENSKI PROGRAMI SVI vrsta — PROGRAMSKA UPUTSTVA** bez kojih se ne može **NANOVUJE**

- LASER GENIUS — Oasis
  - LASER BASIC — Oasis
  - MACHINE LIGHTNING — Oasis
  - THE WRITER — Softtechnics
  - LAST WORD — Myrindomsoft
  - OFFICE MASTER — Gemini
  - GARANCIJA ZA SVE VRSTE USLUGA
- MILIVOJEVI LJUBIŠA**, Petra Laskovića 57, 11000 BEOGRAD, tel: 011/558-007, pošte 17h 037046

**MAXSOFT** najnovije igre i uslužne programe po najpovoljnijim cenama uz najbolje kvalitete. Možete nabaviti kod nas. **DORDEVIC ALEKSANDAR**, Save Kovačevića 42, 11000 Beograd, tel: 011/452-040 ili 451-197 041043



**Zx Spectrum**, nabavite komplet od 12 izabranih sportskih igara. Programi su kompletno sportskih igara su: **Hardball, WS Basketball, Ping Pong, Bump Set Spica, Konami's Tennis, Winter Games I i II, Mc Gulgan Boxing, World Cup Carnival, Hyper sports, International Rugby i Match Day**. Cena kompleta je 1200 dinara plus cena kasete i poštarine. Brza isporuka i kvalitetni snimci. **Maric Miloš, Ustanička 126, 11000 Beograd, Tel. 011/4888-782 031053**

Spectrumovci, za sve one koji su pošli da ozbiljno koriste svoj računar pripremili smo četiri kompleta uslužnih programa. To će te naći veliki izbor od preko 100 uslužnih programa sa mnogo detaljnih uputstava za korišćenje. Nuručite naš besplatni katalog, a kome će te naći spisak uslužnih programa i spisak uputstava koje vam nudimo. Svi uslužni kao i ostali programi se mogu naručiti i pojedinačno. **Maric Miloš, Ustanička 126, 11000 Beograd, Tel. 011/4888-782 031055**



**ZX Spectrum**, nabavite komplet od 13 programa borilačkih vještina. Programi na ovom kompletu su: **Way of the Tiger (3 programa), Amazon Warrior, Ninja Master, Kung Fu Master, International Karate, Gladiator, Sai Combat, Exploding Fist, Kai Temple, Shaolin's Road**. Cena kompleta je 1200 dinara plus cena kasete i poštarine. Osim toga kod nas možete dobiti i dobro poznate kompletne proširih (17 programa) i šah (18) programe. **SPECIJALNI POPUST: NA TRUJENI NARUČENA KOMPLETA ČETVRTI KOMPLET DOBIJATE BESPLATNO!!!** Brza isporuka i kvalitetni snimci. **Maric Miloš, Ustanička 126, 11000 Beograd, Tel. 011/4888-782 031056**

Spectrumovci, ako ne možete da se odlučite prilikom izbora novih programa komplet ratnih igara je prava stvar za vas. Na kompletni ratnih igara ima 12 programa a to su: **Commando, Rambo, 1942, Lightforce, Stallone, Gato, W.A.R.** Brza isporuka i garantovano kvalitetni snimci. Uverite se, **Maric Miloš Ustanička 126, 11000 Beograd, Tel. 011/4888-782 031054**

**SPECTRUM HARDVER**  
Veliki izbor dodatnih uređaja vrhunsko kvalitete po povoljnijim cijenama. Specijalni popust na komplet Kepton interface + Redotisk palica. Informacija: **068/589-987, P. N. P., electronic, Jereveta 12, 56000 Split 031045**

**SPECTRUMOVCI!** Još uvijek vam nudimo programe, pojedinačno ili u kompletima. Besplatni katalog. **Javite se.**  
**Josip Gulić, Bulevar AVNOJ-a 117/3, 11070 Novi Beograd, tel. 011/146-173 031016**

**Zx Spectrum**, najbolja ponuda sortiranih programa na našem tržištu. Nabavite komplet od 13 najboljih simulacija (auto-moto trike, simulatori vožnje aviona, bombardera, helikoptera...) Programi na ovom kompletu su: **TT Racer, Nightmare Rally, Knight Rider, Spitfire 40, Speed King II, Sky Fox, Tomahawk, Rally Driver, Dambusters, Grand Prix Driver, ACE, Turbo Esprit, Rescue on Fractalus**. Cena kompleta je 1200 dinara plus cena kasete i poštarine. Brza isporuka i kvalitetni snimci. **Maric Miloš, Ustanička 126, 11000 Beograd, Tel. 011/4888-782 031052**

## KOMODORE

**COMMODORE 64: KOMPLET 2: IT'S KNOCKOUT, SCOOBY DOO, GALIVIAN, YIE AR KUNG FU 2, WILD WEST 2, BAZOOKA BILL, COMMANDO 2, GLIDER RIDER, RED MAX, IOTH FRAME, 180, LOADERBOARD 3, SPY VS SPY 3, MADONNA, BOULDER DASH 10, LE-GIONS OF DEATH, SEVEN ELEVEN, HIGHWAY ENCOUNTERED, CHOPFLIFER 2, SURFCHAMP, BUNDES LIGA, BOGGIT, SANXION, RUMBLE, FARM PORN0, PORN0 5, POLICE ACADEMY, OMEGA MISSION. KOMPLET=1500 din+KASETA.**  
**TORLO OLIVER, BRAČE ĐUKIĆ 17, 88000 MOSTAR, tel. (088) 34-516 032003**

**ŠAH-ŠAH-ŠAH** osam najboljih programa (tridimensionalnih) sa uputstvima i kasetom (C-60) 3.000.— din. Isporuka odmah pouzdom.  
**Adresa: C-64 Soft, KRČKA 20 C, 41000 ZAGREB 032004**



**IGRE I IZDUBITELNI PROGRAMI NA KASETI I DISKU — besplatni katalog**  
**Stojanović Zlatimirović, Fah 9, 34000 KRAGUJEVAC**  
**COMMODORE 64: KOMPLET 3: CONAN 21 CENTURY, PAPER BOY, TARZAN, FLASH GORDON, LIGHT FORCE, SIGMA SEVEN, ZUB, SPY TROCK, WALL OF SOUND 2, INCREDIBLE, FRUSTRATION, FRUS, MUSIC DEMO 5, KNOTIE INC, BALAKON RAIDER, AVANGER SPECIAL, THE SENTINEL, SUPER MUSIC, FEARLES FRAME, DEFCOM, BILLY DE POST-MAN, 180 RED SUPPER, KEYLETH, X-NO, BACK TO REAL, EREBUS, CAMELOT WARRIORS, BREAKTHRU, SEVEN UP INTRO. KOMPLET=1500 din+KASETA.**  
**TORLO OLIVER, BRAČE ĐUKIĆ 17, 88000 MOSTAR, tel. (088) 34-516 032005**

## kemosoft 072-874-441

Besplatni katalog. Cene 50-100 N.D. Snimam GOST programe, uz brzu isporuku. Od Januara imam: **COBRA STALLONE, WEST BANK, 1943I, FI-RELOAD, HEARTLAND...** itd. Cena +kvalitet=SUPER!!!  
**Željivić Kemal — Kosorica 13 72220 Zavidovići 032001**

**C 64 KORISNIČKI PROGRAMI** sa uputstvima: **SUPERBASE, PLATINE, YU-VIZAWRITE, NEWSROOM, ART STUDIO, ELEKTROMAT, HSG (High Speed Graphics), PRACTICALC, MAE I, PRINT SHOP, COM in i dr.** Katalog i opisom isporuka u toku 24 h. Tel. 047/38-743. Uz drugarski pozdrav, **Prugnić Čedomir, Kragujevačka 8d, 47000 KARLOVAC 032006**

**COMMODORE 64 SUPERHTVO: KOMPLET: 11000 DIN. MIKIE (III), BREAKTHRU, SIGMA 7, PAPERBOY, HIGHWAY ENCOUNTER, SCOOBYDOO, LIGHT FORCE, TARZAN, CAD 64, FLASH GORDON, YIEAR KUNG FU 2, AVENGER, GALIVIAN, 1942 II, DRUIDS, WORLD GAMES (8 DELOVA), HIGHLANDER 1-3, COBAN, THAIBOX 1-3, 21 CENTURU, ČOBANOV BRANISLAV, P. DRAPSINA 53/1, 21480 SRBROBRAN, TEL: 021/730-364 032025**

**MOVIE MONSTER** — Najnovija disketna igra za C64. Izaberite monstra i kreirate da uništavate grove, tenkove, avione i ljude!!! Program + disketa = 3000.—  
**Tel: 021/611-803. 032010**

**Kompiuter Commodore Vic-20+kasetom (novi) i joystick (novi) +program+literatura=besplati mijenjan za flajm disk 1541. Ili sve navedeno programom. Isaković Denis, M. bžduž 33/1, 70000 SARAJEVO, tel. (071) 212-366 032007**

**COMMODORE 64-KOMPLET X: TOP GUN, LIGHT FORCE, XEVIOUS, MARBLE MADNESS, SUPER CYCLES, STARBUCKE 2, DANDY, KETTLER, BLAZER 2, MAD NURSE-PROGRAMI+KASETA=1400,00 DINARA. TRIVKOVIĆ MILOŠ, TRG ANTE BEOGIĆEVIĆA 1/1, 15300 LOZNICA, TEL: 015/82-551 032008**

**THOMYSOFT** — dvogodišnje iskustvo, apsolutno najnoviji programi, literatura, povoljne cijene, popusti, besplatni katalog. **PROFESIONALNA USLUGA**  
**Thomysoft, Ljubiša 5, 41040 Zavidovići, 041/256-520 032012**

**COMMODORE 64: KOMPLET 1: SUMMER GAMES 3 (5 IGARA), POLICE ACADEMY, PARTY GIRLS, SWEDISH EPOTICA (2 PORN0 PROGRAMI), MIKIE, FIST 2, PARALLAX, THAI BOXING 1, 2, 3, MIAMI VICE, 1942, SOLIDER ONE, DAVID BOWIE, KNIGHT GAMES (5 DELOVA), DAN DARE, WAR, HOLLYWOOD OR BUST, SINBAD, I, C, U, P, S., ROOM TEN, POWERPLAY, CAULDRON 2, VELOCIPED 2, N, O, M, A, D., KOMPLET=1500 din+KASETA.**  
**TORLO OLIVER, BRAČE ĐUKIĆ 17, 88000 MOSTAR, tel. (088) 34-516 032011**

**Commodore 64 prodajem** najnovije programe na kaseti i disku. Vršimo uvoznice EPP poruka u sve vrste programa. **tel: (021) 839-044 032013**

**INTER SOFTWARE**  
**COMMODORE 64-INTER SOFTWARE** vam predstavlja u svom besplatnom katalogu izbor samo najkvalitetnijih kasetnih programa za vaš računar.  
**Lakić Goran, Rumenička 12/25, 21000 Novi Sad, tel: (021) 332-671 032017**

**COMMODORE 64: KOMPLET OD 40 KORISNIČKI PROGRAMI ZA SAMO 2000.— DIN. POŠTARINA I KASETA URAČUNATI. ISPORUKA POLUČEM. NARUČIBINE NA TELEFON 011/779-152. GORAN 032024**

**L'AFARE VERA CRUJ 1,2,3, SIK AND DESTROY, EUROPE GAMES (4 prog.), HOWARD DUCK 1 i 2, TOMA-HAWK, D.T. SUPERSTEL, TERA-CRISTA, DONKEY KONG 2, WAR OF WIS, STAR SOLDIER...** itd. Komplet 20 programa + kasetna + poštarina = 1500 din.  
**GORAN VILUVOVIĆ BARISA KIDRIČIĆ, 41/9 34000 KRAGUJEVAC, tel: (034) 67-286 032020**

COMMODORE 64, KOMPLET 4: paper copy, floppy disc, coran, alan ford...  
 KOMPLET 3: galvan, highlander—3, sf. cobra, panther, it's knockout...  
 KOMPLET 2: dan dare, sarxion, sinbad  
 KOMPLET 1: mikie, 1942+parallax...  
 KOMPLET 30 DO 35 PROGRAMA + kasete + PTT=3000 din. TEL: 071/537-106 ILI 071/539-543 032014

SUPERMATE GAMES, NAJNOVIJI PROGRAMI ZA KAZETU I DISK. ALIENS, JEEP COMMAND, GAUNTLET, STAR SOLDIERS, FLASH GORDON II I III. FUTURE KNIGHT, JUDGE DREDD, ASSAULT ON SENTINEL...  
 P.J. DANIEL, GREBNUAK 31, (041) 213-271, 41000 ZAGREB 032015

MONKEY COMPUTER DIVISION  
 2000 YOU, STREET MACHINE, X15 ALPHA, DANDY, FOOTBALL OF THE YEARS, HYPER BALL, NEW PARADOX, KARATE CHOP, MARADONA FOOTBALL, ANJUBER komplet 20 programa + kasete 690 + poštarina = 1500 din.  
 GORAN BOŽIČIĆ, BORISA KIDRIČA 39/3 34000 KRAGUJEVAC, TEL: (034) 60-096 032021

C-64: Najnovije igre superjedinici Miami vitez, druce, cobra, bmx, 1942 + 1, knight rider, azs of aces, bazooka bill, mikie, honda race...  
 PTT=1300 dva kompleta = 2000. Možete i sami sastaviti komplet od 25 igara + kasete + PTT=2000. Sveza nerudba = dve poklon igre. Tražite katalog. Desk Atlanta, Milentija Popovića 132, 23000 Zrenjanin, 023/46-676 032016

— CAMEL SOFT — CAMEL SOFT — CAMEL SOFT — CAMEL — KOMODORCI NE PROPUŠTITE PRILIKU DA NABAVITE HITOVE PROGRAMA U KOMPLETIMA I POJEDINAČNO (100 din). USLUGA JE BRZA, TAČNA I KVALITETNA. SUPER—KOMPLET B = 1. MAGNUM FORCE; 2. SUPER CAM; 3. STALLONE COBRA; 4. SPACE HARRIER; 5. TERRA COGNITA; 6. 1943; 7. FAIRLORD; 8. MICRO RITHAM; 9. SKATE ROCK; 10. HIGHLY LANDER; 11. WHO'S TOUR; 12. ZOOKE SEVEN; 13. PET CRATE; 14. SCOOBY DO (+); SUPER KOMPLET B + KASETA = 1200 din. CAMEL SOFT — GEP. DR. MIHAJLA ILIČA 35, 34000 KRAGUJEVAC, TEL. (034) 33-914 032016

ENGLJSKI I II IGRAJTE I VEŽBAJTE KORISNO 30 LEKCIJA NA VAŠEM KOMODORCI. OBA PROGRAMA + KASETA = 1500.- din. OBA PROGRAMA + DISKETA = 2000.- din.  
 M6 SOFT, III BULEVAR 130/190, 11070 NOVI BEGRAD, TEL: 011/146-744 032039

— CAMEL SOFT — CAMEL SOFT — CAMEL SOFT — CAMEL — KOMODORCI NE PROPUŠTITE PRILIKU DA NABAVITE HITOVE PROGRAMA U KOMPLETIMA I POJEDINAČNO (100 din). USLUGA JE BRZA, TAČNA I KVALITETNA. SUPER—KOMPLET A: 1. KARATE SHOP; 2. MAGNUM FORCE; 3. 1943; 4. INFO DROID; 5. W.A.R. II; 6. HYPERBALL; 7. STAR GLIDER; 8. SKY RANGER; 9. BULDOD; 10. V. 29 FIGHTER; 11. FUTURE KNIGHT; 12. FOOTBALL OF THE YEAR; 13. KNUCKLE BUSTER; 14. CIRCUS CIRCUS II; SUPER-KOMPLET A + KASETA = 1200 din.  
 MILAN NOVAKOVIĆ, VELJKA VLAHOVIĆA 3/2/10, 34000 KRAGUJEVAC, TEL. (034) 68-007 032019

KOMODOROVCI Izaberite vlastiti komplet nudi samim najnovijim programima. Cobra, Gobline II, Return to Oz — jesu stare igre. Ništa vas ne košta da nazovete i naručite besplatan katalog. Daniel Nevidal, B. Radićeva 4, 54000 Osijek, tel. 054/53-082. 032022

PRODAJEM PROGRAME ZA O64 KOMPLET 1: PINK PANTER, HONDA RACE, GHOST'N Goblins II, KWAH, AMERICAN CUP SHAMPION, BIZMARK, BUNDEST LIGUE, WILD WEST, AURIGA, COBRA, RAMBO III (PRAVI), LEMONADE, SPY TRAC, HIGH LANDER I, II, III, KOMPLET 1: SEVENLEVEN, LEER VEEPLE, GAME SOUND, MARRY MELODIES, PUNCH, STRIP SUPER, SNOCROPS, SHANPION SHIP BASEBALL, BAZOOKA BILL, GALIVAN, MOLVI VERNER, KTC, SPACE ACE 200, KIKLOP, BMX, 1 KOMPLET + KASETA = 1200 DIN. OBADVA 2200 DIN.  
 DANILOVIĆ DAVE, PRVE BOKEŠKE BRIGADE 23, 85340 HERCEG-NOVI, TEL: 082/43-276 032023

KOMODOROVCI IRONSOFTWARE  
 Vam nudi sve same igre u super-kompletu. Komplet 2: Donkey kong 2, New uridium (novi), Paradirod 2, Futur knight, 1943, Infodroid, Hypa-ball, Storm, X-29 flight, Paperboy 2, Stalona cobra, Tomhawk, Firerod, Howard duck 2, Highway encounter (hit sa spectrums — super!), Replon 3, War 2, W.C., Criot (ispravljen), Camelot warrior, Space harrier, Fungus, Legend of kage, Legions of death, Zone 7-II., Karate chop, Verunkune, 26 programa + kasete = 1400 dinara.  
 DESPOVIĆ MILEN, Milana Zečara 6, 11210 Beograd, tel. 011/712-442 032042

KOMODORCI 16, 116, 116. VELIKI IZBOR NOVIH PROGRAMA. SNIMAMO POKLANJAM SVI PROGRAMI SU BEZ ZAŠTITE IZIGRE DOBJATE POKOVE ZA BESMRTNOSTI.  
 LJUBISAVLJEVIĆ DRAGAN, 3. OKTOBAR 302/6, 19210 BOR, TEL: 030/33-941 032029

GARFIELD SOFT VAM NUDI I STARE PROGRAME ZA C-64 OD 50—500 DIN SA VAŠOM ILI NAŠOM KAZETOM. JE-DAN PAKET 40 IGARA + KASETA = 2000. DIN.  
 BRUNO KIRSCHNER, TRAVNIČKA 2, 41000 ZAGREB, TEL: 041/329-744 032030

BIGSOFT vam nudi najnovije programe za C-64. Snimamo brzo i kvalitetno na diskete i kasete. Katalog besplatan. Nazovite BIGSOFT, A.V. Butnja 129, 41020 Novi Zagreb, tel. 041/676-291 032045

PRODAJEM ZA C-64 20 EROT-SKI I/40 NAJ-PROGRAMA + KASETA + POŠTA samo 2000 din ILI ZAŠTO NE IMATI OBAVI SAJ NAJBOLE 30 NA-KAKVIM CIJENAMA: 300 NAJ-PROGRAMA SAMO 3000.— DINARA  
 072/39-846 032031

KOMODORCI 64/128. PROGRAMI ZA DISK (BLOODQUIN, GAUNTLET, ALIENS, KASETU (DT SUPERSTES (KONČANO), SPACE HARRIER, FIRE-ROD, 1943, TOMAHAWK, HOWARD DUCK) U KOMPLETIMA I POJEDINAČNO, MNOGO I ZA 128 I CPM. KATALOG 150 D.  
 GRUBOR DEJAN, STOJANA JANKOVIĆA 6A, 11090 BEGRAD, TEL: 011/561-519 032032

L-SOFT. Fantastična prilika Programi za COMMODORE 64 po neverjovatno niskim cijenama. 60 minuta snimljenih programima (90 starijih ili 40 najnovijih) za samo 1000. ND, što znači da je cijena jednog programa samo 10-25 ND. Tražite besplatan L-SOFT katalog!!  
 Levak Nenad, Kumićiova 14, 42000 Varaždin, tel. 042-40603 032046

C-64/128  
 Nudimo igre, korišćenje programe i žezke na disketi i kaseti. Diskete postaje. Moćna i razmena.  
 ALAN HALAS, Vijeće VI. SUK-a 93/1 54000 Osijek, tel. 054/44-226 032033

NEW NOW SOFT I DALJE MISLI NA SVI NAJNOVIJI PROGRAMI, IGRE, USLUŽNI PROGRAMI ZA VAŠE COMMODORE 64 DEVIJA I DALJE VAŠE: KATALOG BEZ PLATAN, CENE NISKE, USLUGA KVALITETNA.  
 ĐUŠAN ADAMOVIĆ, TITOGRAĐSKA 6, 21000 NOVI SAD, TEL: 021/366-205 032034

L-SOFT. Zaštitite vaš COMMODORE I RESET TASTER SA SPECIAL-NIM MIKRO SKLOPKAMA. Cijena sa PTT troškovima: 1300. ND. Komplet TRENDA. Narudžite na adresu: Levak Nenad, Kumićiova 14, 42000 Varaždin ili telefon: 042/40603 032047

COMMODORE 64/PC 128  
 KOMPLET 1: FIST II, III, BAZOOKA BILL, SANDSIOK II, BULL-RIDING, TRANS-FORMERS II, USH-MATA, SOULDER ONE, BOULDER DASH 13, B. BEARING, LOG-ROOLLING, MASTERS OF THE UNIVERSE!!! GREAT ESCAPE!!!, S. FORCE HARRIER, PSY III KOMPLET 2: TARNAN, INBAND, RED-MAX...  
 ING. DARKO DIMITROV, DIMITAR BLAGOJEV 9b, 91000 SKOPJE, TEL. 091/415-213 032036

COMMODORE 64: PROFESIONALNI PREVOZI. Svaki trend u profesionalno urađen i brzo ukočen.  
 \* GRAF 64 1000 DIN. \* STAT 64 1000 DIN. \* KONTOAMT 1000 DIN. \* TRIKOVI 64 1000 DIN. \* PRACTICAL 1000 DIN. \* MULTIPLAN 1000 DIN. \* SIMONS BASIC 1300 DIN. \* SUPER BASE 1500 DIN. \* FLIGHT SIMULATION II (sa silikama) 1500 DIN.  
 C-64 PRILUŽUJE TO JE USLAVI KNJIGA prilagođena početnicima i svima onima koji žele da savladaju korektivno boric. Ima 100 strana I DEBE-LE KORICE 3000 din. — Šta će vam dobar program bez dobrog uputstva.  
 MINJA VIŠIŃKI, Jurja Gagarina 141/83, 11070 Novi Beograd, tel. 011/157-756 032050

COMMODORE 64, RAMBO SOFT NUDI NAJNOVIJE SVJETSKETE HITOVE: FLASH GORDON, LIGHT FORCE, BREAK THRU, DEFCON, XENO, BILLY DA POSTMAN, FIGHTER, MAGNUM, KNOTIE, KAY TIEH, SIGMA SEVEN, SKY LAP, TARNAN, ZUBB, BACK TO REAL, BALVA-CON RIDER, HIGH LANDER I, II, X29 FIGHTER, STORM EREBUS, LEADER-BOARD 3, ... BORO KVALITETNO, JEF-TINO + PTT = 1900 DIN.  
 BORIŠ GLAVČIĆ, DANKA MITROVIĆA 14, 71000 SARAJEVO, TEL: 071/451-272 032037

MENJAMO najnovije i odlične stare hitove za C 64. Ako želite da dopunite svoj katalog pišite nam. SMOLA SOFT, Solska 11, 68280 Brestanica, Slovenija 032063

C64 program 50 din. Komplet: rai-ne, borilacke i sportske. Komplet sadrži 10 programa i stajce 500 din. Marković Dragan, Kosanićeva 69, 12000 Polzarevac, 012/25-402 032053

SPORTSKA PROJEKTA NUDIMO VAM U PRVOM SP. — PROGRAMU — RASPORED PRAVNOSTA — TRADICIJA I BILANSI — BIORITAM IGRACA I EKIPA — PUNI USLOVNI SISTEMI SP. + 10 PROGRAMA + KASE + POŠT. = 2090 D ZA SPEKTRUM OD 64. MARTA  
 011/591-791, 649-600 Kuga Branko Borska 92/1 Beograd 032063

C 64: STUDIO-R nudi svim pravim hakera-ma i onima koji žele da to i postanu. mogućnosti samostalnog razbijanja pro-grama. Programme iz naše uputstvo mo-gaže razbijati i ako ne poznajete mašins-ke jeziki! Ubacite sami svoje ime u željeni program! UPUTSTVO + MONITOR-program + kasete = 3000 (tri hiljade) ND! Tel: 083/31-743 JOJIC VLA-DIMIR, Ratka Vujovića Čoče 17, 81400 Nikšić 032076

Komodorovci IRONSOFTWARE vam nudi sve same super igre u superku-per kompletu. Komplet: Paperboy, Tarzan, Flash Gordon, Galvan, Cobra, Shaolin Star, Bismark, SVS 4, Boulderdash 8—12, Vietnam, Ghost'n Gobline 2, Top Gun, Yie-ar kuog-fu 2, Miami dice, Sweden Eric-son, Party Girls, Exploding F.I.S. 2.3, Mad Nurse, It's Knox Out, S.S. Spurt-rik, 25 programa + kasete = 1400 din.  
 Despotičić Milen, Milana Zečara 6, 11210 Beograd, tel. 011/712-442 032069

C 64: Ako ste ozbiljan korisnik Vašeg računara, ne očajavajte! STUDIO-R Vam nudi po uzatno niskim cijenama vrhunske korisničke programe sa uputstvima, i to na kaseti (SAD 64 111) ADDRESS 64, HELP 64 + 111...). Za detaljnije informacije se obratite na tel: 083/31-743 (VLADI-MIR) 032077

COMMODORE HARDVER NOVO — ROM moduli do 32 K, uskoro 64 K. SIMONS Basic II, Okaford paskal, Turbo su samo neki od mnogobrojnih modula koje možemo razvijati kod nas. Veliki izbor dodatnih uređaja. Svijetlosna olovka, printer interface, programator EPROMa, modem, palice, literatura itd.  
 P.N.P. electronic, Jeretova 12, 58000 Split, telefon (058) 589-967 032075

COMMODORE 64: Prodajem pojedinač-no najnovije programe direktno iz me-morije komputera za 50 din. MD 128: kasetni programi 100 din. PROVERITE VELJKOVIĆ MIROSLAV, Rasnina 4, 18000 Niš, tel.: 018/334-539 032084

LJUBILETI SPORTSКИH IGARA I SPORT STUDIO VAM NUDI NAJPOPULARNIJE SPORTSKE IGRE ZA C-64 ZA SAMO 50 DINARAI BESPLATAN KATALOG. TOR-DAJ BRANISAV, ANTONA KLEMENČIĆA 20, 11186 ZEMUN POLJE, TEL: (011) 656-146 (SAMO RADNIM DANIMA OD 10—14 časova) 032085







ZAMIR SOFTI Amstradovci! Ovak mjesec posebno iznenađenje, programe možete nabaviti u kompletnu. Dva najnovija kompleta: KOMPLET 25: Avengers, Dan Dare, Hezenuche, Scooby Doo, Light Force, Harley Headbanger, Colgate, Inti, Karate I, KOMPLET 26: Inti, Karate II, Top Gun, Impossible Mission, Ivory Warriors, The Keyfator, Jack the Nipper, Mool Cresta. Jedan komplet staje na jednu stranu kasete C60. Cijena najnovijih kompleta 1800 + kasete. Stariji kompleti su jeftiniji. Kvalitet zagaranovani. Pored kompleta programi smimamo i pojedinačno (200 dinara). Ne oklijevajte, tražite besplatan katalog sa naznakom za AMSTRAD na adresu: Danijel Kurtović, Maršala Tita 72, 88000 Mostar ili na tel: (088) 53-644 033018

CPC 464: SVI NAJNOVIJI HITOVI U KOMPLETNIMA (SUPER JEVTINO) I POJEDINAČNO (JEVTINO). PROVERITE I NEĆETE ŽALATI! BESPLATAN KATALOG I EXPRESS ISPORUKA. PEJČIČINSKI ZORAN, BEOGRADSKA 47a, 97000 BITOLA, tel. 097-42-241

033012

## ATARI

PRODAJEM programe za ATARI ST po najnižim cenama. HERCEG EMIL, Ratimira Heroega 2, 42000 Varaždin, tel: 042/44-956 pre podne. 034001

Prodajem ATARI ST programe — poslovanje, grafički, utility, igre. Cijene vrlo povoljne. Katalog besplatan. GADINA DUBRAVKO, Krajiška 16/4, 44103 Sisak, tel: 044/34-700 034003

ASCII SOFT: I ovog meseca novi program: ONE ON ONE, KIK START (na kaseti)! Javite se, katalog je besplatan! PANDUROV ZORAN, Đurđevačka 33, 23000 Zrenjanin, tel: 023/63-521 034002

ATARI \* ATARI \* ATARI Veliki izbor najkvalitetnijih programa na kaseti I disketi i literature za Atari 800XL/130XE (više od 300 igara, 100 korisničkih programa, 15 najboljih knjiga ...). Za najnoviji specijalni katalog sa kojim više nema muka (iako usmiriti programe, od kojih otkrivate, koje dužine ... ) sa svim mogućim popisima, objašnjenjima, cenama, ... poslati 100 t.d. LACMANOVIĆ DEJAN, Sindrićeva 31-a, 23000 Zrenjanin, tel: 023/66-879, pošte 14 H. 034004

ATARI MASTER CLUB, XL/XE prodaja—razmena. Katalog 150 din. Jovanov Slobodan, Pivarska 2/A, 23000 Zrenjanin. 034006

ATARI SUPER COPY DARAN COPY 2+kaseta+PTT+uputstva=1400,—din. Kopiraju sve programe za ATARI serije XL, XE. tel: 076/61-130 034005

ATARI ST HARDVER Veliki izbor dodatnih uređaja vrhunске kvalitete po povoljnim cijenama. RAM 1Mb, ROM-ovi, TV-modući, disk 720 kb, Fast Basic kartridge, literatura, programi. Tel: 058/589-987, P.N.P. electrojic Jeretova 12, 58000 Split 034007

## HARDVER

Prodajem novi, neraspakovanu profesionalnu tastaturu TREND sa reset tipkom. Cena 30.000 din. GOLUBOVIĆ BOBAN, Karađorđeva 46/59, 11300 Smederevo, tel: 026/26—184 036001

Prodajem sledeće KIT KOMPLETE za ZX SPECTRUM: PROGRAMATOR EPROMA, prema šemi u Računarima 2 (35.000 din.), A/D PRETVARAČ prema Moj Mikro, apr 86 (30.000 din.), SPICA HARDWARE, KERŠKINOVA 4, 61000 LJUBLJANA. 036003

PRODAJEM SPECTRUM 48K PLUS TREND TASTATURU. Vikendom tel: 022/425-432 036002

AMSTRAD CPC 464 s monitorom, s programima I literaturom prodajem. Tel: 033/53-383 036004

Popravljam kućne računare tipa Komodore, Spectrum, Amretad. MARKOVIĆ DRAGAN, Kovanlučka 56, 18000 Niš, tel: 018/42-028 036005

## Literatura

Spectrums sami možete popraviti samo uz fotokopiju knjige „Spectrum service manual“ (3000 din.). Sve o Vašem Spectrumu i njegovim hardverskim dodacima saznajte iz fotokopije knjige „Spectrum hardver manual“ (4000 din.). S obje fotokopije besplatno dobivate adresu na kojoj možete naručiti dijelove. Adresa: Sušter Borde, Neškovićina 13, 58000 SPLIT 037002



LITERATURA Za sve koji se ozbiljno bave računarima neophodna je i dobra knjiga. COMET software Vam nudi originalnu LITERATURU na ENGLESKOM jeziku — knjige za vaš računar ZX SPECTRUM QL COMMODORE. BBC, APPLE — Mikroprocesori — Programski jezici — Operativni sistemi — Obrada teksta — Servisni priručnici GARANCIJA ZA SVE VRSTE USLUGA MILOVANOVIĆ LJUBISA, Petra Lekovića 57, 11300 BEOGRAD, tel: 011/558007 pošte 17 H. 037003

### Originalni jeftiniji od foto-kopije

Originalna literatura po nabavnim cenama za spektar, komodor, 68000 ...

MC6800 16/32 bit Microprocessor Programmers Reference Manual	10.000 din
Self-Guided Tour Trough 68000	7.500 din
The ADA Programming Language	11.000 din
The CPM/H Handbook with MP/M	9.000 din
8086/88 Assembly Language Programming	11.000 din
Tutorial Pascal for BASIC Programmers	8.500 din
MS DOS User's Guide	10.000 din
Masterint Symphony	10.000 din
Using Enable	10.000 din
Mastering WordStar	9.000 din
Macintosh: Programmers Reference Guide	12.000 din
Apple Ite Programmers Reference Guide i još preko pedeset naslova	12.000 din

Sve knjige su izdate 1985. I 1986. godine, imaju između 300 i 800 stranica potpuno su nove. Požurite, količine su krajnje ograničene — jedan primerak naslovu. Besplatan katalog. M. Mastilović, Driška 2, 11080 Zemun

## edicija ZOROASTER!

Ako zaista volite kompjutere, evo pravih knjiga za vas!

### 1. Vilijem Gibson: NEUROMANSER

Roman koji je za prošlu godinu dobio sve SF nagrade u svetu (HUGO, NEBULA, LOKUIS, FILIP K. DIK) i time postao najnagrađivaniji roman u istoriji naučne fantastike. Govori o hakerima posebne vrste, koji proljavaju u programe snabdevane ubistvenim zaklintonim signalima. Izuzetna napetost radnje i superioran stil autora ni sa neće ostaviti ravnodušnim! Knjiga izzila u februaru 1987. Cena 2500 d.

### 2. Daglas Adams: VODIČ KROZ GALAKSIJU ZA AUTO-STOPERAE

Po ovoj svetski poznatoj trilogiji napravljene je najpopularnija kompjuterska igra u svetu ove godine. Sada se i čitaocima u Jugoslaviji pruža šansa da osete fenomenalni Adamsov humor. Ova knjiga vas neće samo zabaviti, ona će vas zasmejati od suza! Knjiga se odmah isporučuje. Cena 3300 dinara.

Preko ovog oglasa možete nabaviti i ostala izdanja edicije ZOROASTER:

### 3. Isak Asimov: BOGOVI LIČNO

Roman za koji je veliki Asimov dobio nagrade HUGO i NEBULA. Knjiga se isporučuje odmah. Cena 2200 dinara.

### 4. Džeims Bilis: LETEĆI GRADOVI

Jedna od najpoznatijih serija u naučnoj fantastici. Knjiga se odmah isporučuje. Cena 3000 dinara.

### 5. Ursula LeGuln: NEBESKI STRUG

U bogatom opusu najvećeg majstora SF-a, našao se i ovaj izvanredni roman o prokletstvu snova. Knjiga se isporučuje odmah. Cena 1500 dinara.

Knjige ediciji ZOROASTER platite pouzedećem.

### NARUĐBENICA

Ovim naručujem knjige pod brojevima 1 2 3 4 5

Ime i prezime .....

Ulica i broj .....

Poštanski broj i mesto .....

(Knjige ću platiti poštaru)

Naruđbenicu šalite na adresu: Edicija ZOROASTER, Branislav Brkić, 11070 Novi Beograd, poštanski fak 22

PREVODI  
Sprekovrhvatni, latinica:  
1. PROGRAMSKI JEZIK C.

4800,00 din.

2. ATARI ST .....  
— Priručnik ..... 1800,00 din.  
— BASIC ST ..... 2400,00 din.  
— LOGO ST ..... 1400,00 din.  
više 500 din. za post. isporuku pouzedećem.  
M. Karababić, Post restant, 19210  
— Bor ..... 037001

## IBM PC

IBM PC/XT i kompatibilni računari. Profesionalno prevedena uputstva za programe: dBASE II, dBASE III, LOTUS 1-2-3, MS DOS 3.2, GW BASIC. Ponuda originalnih uputstava i programa. Obavještenje o naruđbi na telefon: 071/621-025 ili 071/455-562

NAJJEFTINJI SEX KOMPLET na tržištu. Spectrum 18 programa = 1996,— din. Komodor 30 programa. 2.199,— din. Uračunata cena kasete. Ekspres isporuka. Kvalitet i uzbujanje garantuje. BROTHERS SOFT, Ivo Lole Ribara, 67. 25272 Bački Monoštor

## RAZNO

PRODAJEM ili menjam video kompjuter sistem ATARI 2600 za ZX Spectrum ili Commodore 64. Javite se na adresi: FEKETI ISTVAN, Petefi Sandora 58, 26214 Debeljaca 038003

Na tržištu postoje tri verzije paketa „Laser“ koje se razlikuju u nekoliko sitnica. Prva verzija je namijenjena za računare „amstrad“/„sneider CPC 464“, CPC 664 i CPC 6128, druga za „spektrum 48“ i „spektrum+“, a treća računarnima „spektrum 128“ i „spektrum+2“. Razlike između pojedinih verzija biće spomenute u tekstu. Inače, autor se skoro stopostotno vjerovatno pretpostavlja da je program „Magus“ (koji je evidentiran u skoro svim katalogima programa za „spektrum 128“) upravo treća verzija „Lasera“.

U ovom broju „Računara“ biće govora o editoru/asembleru „Toolkit“, i pseudokompajleru „Feniks“ dok ćemo u idućem pisati o monitoru i onom što ih uz njega.

Nakon što otkucamo LOAD, dočekuje nas oporni bežik loader (preko 3K) koji nas pita da li želimo kopiju programa na mikrodravju (sva šta se u ovom tekstu odnosi na mikrodravj važi samo za „spektrum“, a ono što se odnosi na disk za „amstrad“, da li želimo promijeniti boje ekrana i promjenu tipa printera, te da li želimo da se sa asemblerom učita i „Toolkit“ ili pseudokompajler „Feniks“ („Hash extension“) koji ne radi ako assembler nije u memoriji.

Nakon ovih dosadnih pitanja, pričekaćemo koju sekundu ako imamo mikrodravj ili disk, ili koju (joj užasaj) minutu ako imamo kasetofon. Uz put, najveća mana „Lasera“ je upravo učitaivanje.

Druga (i posljednja) mana programa je zauzete memorije (ova mana, naravno nije prisutna na „amstradu 6128“ ili „spektrumu 128“). Naime assembler zauzima čak 23K. Još ako dodamo „Toolkit“ (1.5K), „Feniks“ (3K), koji, srećom, ne moraju biti prisutni, te razne bafere, ispada da ostaje premalo memorije za tekst i objektni kod. Srećom, to ne dolazi do izražaja u pretrajanjoj formi zbog toga što je tekst (osvojeno) tokenizovan u gotovo nevjerojatno kompaktnu formu (6–7 puta je kraći od GENSP-ovog) i zbog odlično urađene INCLUDE/OPENOUT komandi (po gotovo za mikrodravj ili disk). Uz put verzija za „spektrum 128“ i +2 zauzima 32K zbog toga što se „Toolkit“ i „Feniks“ uvijek učitaivaju zajedno sa asemblerom i zbog nekih novih komandi. Pomenuta verzija se učitava u RAM disk i radi iz RAM diska tako da upadne ne tek „normalna“ memoriju, izvorni tekst se takođe smješta u RAM disk.

### Editor nad editorima

Nakon (mikro)trpnog učitaivanja, bežik loader se briše, a RAMTOP postavlja na 25500 (za „spektrum“).

### 50/komercijalni softver

Na samom početku dočekuje nas odličan ekranski editor (ubjedljivo najbolji koji sam ikad vidio). Radi sa 42 (40 na „amstradu“) znaka u redu i koristi specijalni bafar za smještanje teksta koji se kuca. Inicijalno je taj bafar postavljen na 01vkih 10K, ali se pomoću specijalne naredbe veličina tog bafara može mijenjati i po potrebi smanjiti na samo 1K (čime dobijamo znatno veći prostor za izvorni i objektni kod). Editorski bafar služi za smještanje bilo čega što kucamo (ne za sors fajl). Ekran se poneša kao prozor unutar bafara — ono što nam pobjegne sa ekrana možemo vratiti prostim pomicanjem kursora.

Urti je važno primijetiti da editor dozvoljava linije koje sadrže više asblerskih instrukcija. Ako otkucamo 10 loop:DEC BC (ENTER) zatim LD AB (ENTER), OR C (ENTER) i JR NZ,loop (ENTER), i nakon toga 20 RET (ENTER) listing će izgledati ovako:

```
10 loop: DEC BC
        LD AB
        OR C
        JR NZ,loop
```

```
20      RET
```

Zbog toga, AUTO komanda nije ni potrebna. Jedna linija može biti dugačka i desetak ekrana. Takođe, ne postoji ni tipka za tabulaciju, jer se tabulacija obavlja automatski. Maksimalna dužina labela je 240 znakova. U imenu labela može se nalaziti znak „.“ ili „\$“, ime labela obavezno završava znakom „:“ koji služi samo kao separator i nije sastavni dio imena. Komentar se i dalje odvajaju znakom „;“.

Za editovanje stoje nam na raspolaganju slijedeće komande (oznake u zagradama odnose se na „amstrad“):

```
strelce
CAPS 4 (SHIFT 9)
CAPS 9 (TAB ili SHIFT RIGHT)
CAPS 0 (FLR)
SYM 0 (CTRL)
CAPS 3 (SHIFT DEL)
SYM F (CTRL DEL)
SYM I (SHIFT TAB)
SYM W (COPY)
CAPS 1 (CTRL TAB)
SYM G (CTRL „L“)
```

Na „amstradu“ postoje i specifična pomjeranja kursora koja se dobiju pomoću SHIFT ili CTRL i strelca. Na „spektrumu“ su zanimljive kombinacije SYM Q i SYM E. Tipka SYM A (ESC) služi za prekid bilo koje operacije.

Unos svake instrukcije praćen je sintaksonom kontrolom i odgovarajućim komentarom. Npr. otkucate li LD H,BC) dobićete poruku „illegal second operand“ i kursor će se postaviti iznad za zarez, tj. na mjesto greške. Nakon što uklonite sintakson grešku, poruka će nestati sa ekrana. Assembler će vam objasniti

vrlo detaljno bilo koji tip sintaksne greške.

### Specifičnosti assemblera

Assembler je standardni dvoprolazni makroassembler pravljen po svim Zilogovim standardima. Numerički parametri svih instrukcija Z80 procesora (npr. LD BC,nnnn) mogu biti kako konstante, tako i izrazi. Konstante mogu biti dekadni, (npr. 239), binarni (%101101), oktalni ( 7326) i heksadecimalni brojevi, koji se mogu pisati na dva načina (# \$C3A ili \$C3AH), te azbucne konstante („A“). Izrazi mogu sadržavati konstante, labela, te brojač lokacija (\$) i adresa (.) koji će biti kasnije objašnjeni. U izrazima su dopušteni slijedeći operatori (broj označava prioritet):

### Binarni operatori

+	(sabiranje, 7)	-	(oduzimanje, 7)
*	(množenje, 8)	/	(cjelobrojno dijeljenje, 8)
<	(ostatak dijeljenja-MOD, 8)	>	(veće od, 8)
%	(manje od, 5)	=	(veće ili jednako, 5)
<=	(manje ili jednako, 5)	?=	(jednako, 4)
!=	(nije jednako, 4)	<<	(pomerenje ulijevo, 6)
>>	(pomerenje udesno, 6)	&	(rotacija ulijevo, 6)
↔	(rotacija udesno, 6)	∩	(binarni AND, 3)
	(binarni OR, 3)	⊕	(binarni XOR, 3)
∩∩	(logički AND, 2)	⊕⊕	(logički OR, 2)

### Unarni operatori

-	(unarni minus, 9)	!	(logički NOT, 9)
∩	(binarni NOT-CPL, 9)	!	(16-bitni sadržaj adrese, 9)

Dobar poznavalac programskih jezika lako može primijetiti da je mnogo operatora pozajmljeno iz jezika C. Očito je da Laser, za razliku od GENSA-a, poštuje priori-

Ova dva brojača obično imaju istu vrijednost, ali ne mora uvijek tako da bude.

Svaka greška uočena prilikom asblerivanja ispisuje se punim tekstom (npr. „number out of range -128,127“), a ne samo „ERROR 10“, kao kod GENSA-a. Postoji ukupno trideset i dve vrste sintaksonih i trideset vrsta trunih grešaka poredanih po težini u tri klase: „warning“, „error“ i „fatal error“.

### Makro naredbe

Makroinstrukcije predstavljaju prilično jako oružje GENIUS assemblera. „Laserove“ makroinstrukcije nisu tako moćne kao u „Machine Lightingu“, ali su itekako upotrebljive. Definišu se pomoću naredbe MACRO u sintaksi labela: MACRO parametri. Svi parametri moraju

počinjati znakom „“ (backslash). Slijede dva primjera koja će najbolje pokazati definisanje makronaredbi:

```

outchr: MACRO regl, RD A, code
        LD A, code
        RST 16
        ENDM

swap: MACRO regl, reg2
        PUSH regl
        PUSH reg2
        POP reg1
        POP reg2
        ENDM
```

Ova makronaredba štampa dati karakter. Slijedeća makronaredba zamjenjuje vrijednost dva registra (doduše na krajnje neelegantan način):

```

swap: MACRO regl, reg2
        PUSH regl
        PUSH reg2
        POP reg1
        POP reg2
        ENDM
```

Definisane makronaredbe možemo koristiti u svom programu na slijedeći način: outchr 85 ili outchr „A“+1 ili swap HL,BC. Očito je da se kao parametri makronaredbe mogu upotrebiti i izraz, ali konstrukcije tipa LD A, code+1 u samoj definiciji makronaredbe nisu dozvoljene.



Da li smatrate da popularni GENS ne zasluhuje ni prolaznu ocjenu, prvenstveno zbog svog blago rečeno užasnog editora? Da li ste pomislili da je takođe dobro poznati debager MONS pogodniji za lovljenje nedevda nego bagova? Ako je odgovor na ova dva pitanja pozitivan, rješenje je vrlo jednostavno: bacite GENS i MONS u kantu za smeće i nabavite Laser Genius . . . programski paket koji obuhvata više programa za jednostavan rad u mašinskom jeziku: editor/sembler, pseudokompajler „Feniks“, assembler „Toolkit“, disassembler/monitor, RPN analizer i još neke sitnice. Paket je djelo programera firme Oasis Software koja je već ušla u legendu sa svojim programima „Machine Lighting“, „White Lighting“ i „Laser Basic“.

### Direktne naredbe editora

CLS	Briše ekran i editorski bafer.	TAPE	Proglašava kasetofon za izlazno-ulaznu jedinicu.
SETSPACE n	Određuje veličinu radnog prostora editora u bajtovima. Na „spektrumu 128“ ova naredba nije uvedena zbog dovoljno velike memorije.	TAPE.IN	Proglašava kasetofon za ulaznu jedinicu.
LIST [b]	Lista program ili njegov dio.	TAPE.OUT	Proglašava kasetofon za izlaznu jedinicu.
LLIST [b]	Kao LIST, ali sa ispisom na štampaču.	MDRV [n]	Proglašava mikrodrajv broj n za ulazno-izlaznu jedinicu (default n=1). Na „amstradu“ se koristi naredbe DRIVEA i DRIVEB za biranje default drajva, te naredba DISC koja proglašava disk za ulazno-izlaznu jedinicu.
DELETE [b]	Briše blok linija.	MDRV.IN	Proglašava mikrodrajv za ulaznu jedinicu (na amstradu DISC.IN).
COPY b1,b2	Kopira blok linija b1 na poziciju b2.	MDRV.OUT	Proglašava mikrodrajv za izlaznu jedinicu (na amstradu DISC.OUT).
MOVE b1,b2	Premješta blok linija b1 na poziciju b2	REN x\$,y\$	Mijenja ime programu x\$ u y\$ na disku (samo „amstrad“).
RENUM [n1,[n2],[n3]]	Prenumeriše program tako da početna linija bude n1, a step n2. Prenumeracija se obavlja od linije n3 (default n1=n2=10, n3=0).	ASSEM	Asembliira program
FIND x\$(,b)	Traži string x\$ u bloku linija b.	ASSEMBC	Asembliira program bez brisanja vrijednosti labela koje su ostale od prethodnog asembleriranja.
REPLACE x\$,y\$(,b)	Zamjenjuje string x\$ stringom y\$ u bloku linija b. Zamjena se vrši po potrebi po principu „replace (y/n)?“.	TABLE [n]	Štampa imena i vrijednosti labela u ASCII poretku. Parametar n određuje format outputa (default n=16)
NEXT	Nastavlja izvođenje FIND ili REPLACE komande ako je bila prekinuta.	LTABLE [n]	Kao TABLE, ali sa ispisom na printer.
PRINT x[,n]	Štampa vrijednost izraza x u bazi n u predznačajnoj notaciji (default n=16)	TABLEN [n]	Štampa imena i vrijednosti labela u numeričkom poretku.
UPRINT x[,n]	Štampa vrijednost izraza x u bazi n u nepredznačajnoj notaciji (default n=16).	LTABLEN [n]	Kao TABLEN, ali s ispisom na printer.
BASE n	Postavlja default bazu za PRINT i UPRINT naredbe. Moguće vrijednosti za n su 2,8,10 i 16.	MISSING [n]	Štampa imena svih labela koje su ostale nedefinirane prilikom asembleriranja.
FORM	Šalje „form feed“ znak na printer.	LMISSING [n]	Kao MISSING, ali sa ispisom na printer.
WIDTH [n]	Određuje širinu stranice na štampaču (default 65536)	UNUSED [n]	Štampa imena i vrijednosti svih labela koje su definirane u programu, ali nigdje nisu korištene
LENGTH [n]	Određuje broj redova na stranici (default n=65536)	LUNUSED [n]	Kao UNUSED, ali sa ispisom na printer.
MARGIN [n]	Postavlja lijevu marginu za štampač (default n=0).	CLEAR	Briše tabelu sa imenima i vrijednostima labela.
SAVE x\$(,b)]	Snima blok linija b na disk, kasetofon ili mikrodrajv pod imenom x\$. Tekst se snima u blokovima po 2K, te je pogodan za INCLUDE komandu (default x\$=ime zadnjeg učitano g fajla)	EXECUTE x[,p1,p2,...]	Starta mašinski program koji počinje na adresi x. Opcioni parametri p1,p2,... mogu biti bilo kog tipa (npr. EXECUTE start, „računari“=5,„10110“, „prog“) i adresiraju se registrom IX. Registar A sadrži broj parametara. Kod „spektruma 128“ postoji i parametar koji određuje željenu stranicu RAM-a.
LOAD x\$(,b)	Učitava program x\$ i smješta ga na poziciju b.	STATS	Prikazuje mapu memorije (gdje se nalazi assembler, sors fajl, baferi, tabela labela, itd.). Na „spektrumu 128“ daje je i mapa RAM diska.
VERIFY x\$(,b)	Verificira program ili njegov dio.	EXIT	Izlaz u bejzick (povratak u assembler vrši se pomoću RANDOMIZE USR 65533).
CODE x\$,x1,x2	Snima dio memorije od adrese x1 do adrese x2 pod imenom x\$. Kod „spektruma 128“ postoji i četvrti opcioni parametar koji bira odgovarajući blok RAM memorije.	HOUSEWORK	Poziva ugrađeni COPY/SEARCH program (koji je sastavljen dio assemblera) i služi za kopiranje fajlova proizvedenih pomoću „Lasera“. Posjeduje desetak svojih naredbi, od kojih je najvažnija COPY x\$,y\$ pomoću koje se kopiraju fajlovi. Povratak u editor vrši se pomoću EXIT.
CAT	Prikazuje katalog programa na disku ili mikrodrajvu (na „amstradu“ CAT radi i sa kasetofonom).		
ERA x\$	Briše zapis x\$ sa mikrodrajva ili diska		

Parametri u srednjim zagradama mogu se izostaviti. U popisu naredbi x\$,y\$,... označava proizvoljan string, n,n1,n2... označava decimalne 51/laser genius

konstante, x1,x2,... označava proizvoljan izraz, a b,b1,b2,... označava blok linija u „komodor“ stilu (npr. 10-50 ili -50, ili 10-, ili 30, itd.). Ukoliko se izostavi blok linija, podrazumijeva se blok 0-65534.

## Naredbe „Toolkita“

LOADASCI  
x\$[,n1],[n2]]

Učitava i tokenizuje tekst koji nije nastao pomoću „Lasera“; x\$ je ime fajla, n1 opcija tokenizacije (default 0), a n2 broj naredbi koje će biti smještene u jednu liniju (default 10). Opcija može biti od 0 do 15 i svaki bit opcije (ukupno 4) ima posebnu funkciju (za tokenizaciju tekstova „spektrumnog“ GEN5-a treba staviti n1=12, dok je za „amatrado“ GEN5 n1=8). Sintaksno neispravna linija biće prilikom tokenizacije označena sa „bad line“.

ASSEMBL

Vrlo komplikovana komanda — vrši selektivno asembliranje izabranih rutina iz vaše biblioteke programa; postupak je u uputstvu vrlo detaljno opisan na pune dve strane.

EXPORT x\$

Snima tabelu sa imenima i vrijednostima labela na traku, disk ili mikrodrav.

IMPORT x\$

Učitava tabelu labela (snimljenu pomoću EXPORT).

REDUCE

Radi slično kao CLEAR, ali ne briše one labela koje su zaštićene pseudonaredbom asemblera CARGO.

Direktne naredbe editora koje rade samo ako je „Toolkit“ prisutan u memoriji.

## Pseudonaredbe asemblera

ORG x

Postavlja brojač lokacija (\$) na adresu x. Tako asemblirani program radiće ispravno samo na adresi x (default \$=RAMTOP).

Put x

Postavlja memorijski brojač (!) na adresu x. Objektni kod biće smješten na adresu x, što ne znači da će se on ispravno izvršavati na toj adresi. Možemo staviti PUT \$ i tada će memorijski brojač dobiti istu vrijednost kao i brojač lokacija (\$). Default := RAMTOP. Kod „spektruma 128“ može se zadati i željena stranica RAM-a.

DEFB parametri

Smješta bajtove u objektni kod. Za razliku od GEN5-a, moguće je mijenjati riječi i brojeve, npr. DEFB 22, „RACUNARI“, 1987. Naredba DEFB može se skraćeno pisati kao DB.

DEFM parametri

Naredba uvedena samo zbog kompatibilnosti sa GEN5-om. Isti efekat se može ostvariti pomoću DEFB.

DEFW parametri

Smješta 16-bitne brojeve u objektni kod. Može se skraćeno pisati kao DW.

DEFS x

Rezerviše prostor od x bajtova u objektnom kodu (skraćeno DS).

labela: EQU x

Definiše broju vrijednosti labela.

labela: DEFL x

Mijenja vrijednost već definisane labela.

Jedan primjer dovoljno pokazuje:  
count: EQU 1  
\*WHILE count<=15  
DEFB 2\*count-1  
count: DEFL count +1  
\*ENDW

Ova sekvenca pravi tabelu prvih 15 neparnih brojeva koristeći WHILE ... ENDW petlju. Naredba DEFL može se skratiti na DL.

COND X...

... ELSE ...

... .ENDC

Ukoliko je uslov x ispunjen, asembliranje se nastavlja. U suprotnom asembliranje se nastavlja od pseudonaredbe ELSE. Kraj sekvence za uslovno asembliranje označava se sa ENDC. ELSE instrukcija može se izostaviti. U tom slučaju koristi se samo COND ... ENDC konstrukcija. Konstrukcija COND X ... ELSE ... ENDO donosi identična konstrukcija IF x ... ELSE ... END iz GEN5-a.

labela: CARGO

Služi da zaštiti labelu od brisanja pomoću REDUCE komande „Toolkita“.

## Pseudokompajler „Feniks“

Doli smo do najzanimljivijeg dodatka asemblera „Laser“ — feniks pseudokompajlera „Feniks“. Naime, riječ je o kompajleru vrlo jednostavnog programskog jezika — feniks (pošto autor prvi put čuje za ovaj jezik, pretpostavlja da je u pitnju ploid mašte OASIS-ovih programera). Jezik je relativno malih

mogućnosti, ali se njegove instrukcije mogu mijesati sa instrukcijama Z80 procesora, što daje ogromne mogućnosti. Zajedno sa „Laserom“ dobivate demoo programe za nevjerojatno brzo crtanje krugova i elipsi i za traženje prostih brojeva. Demo programi su pisani u dve verzije: u čistom „mašincu“ i u kombinaciji 90% fenik 10% asembler. Ova druga verzija po brzini gotovo ništa ne zaostaje od prve, a dužina objektnog koda je svega

## Pseudoopcije asemblera

\*SCREEN I

Uključuje ili isključuje bilo kakav pristup ekranu za vrijeme asembliranja (default ON).

\*PRINTER I

Uključuje ili isključuje izlaz na printer za vrijeme asembliranja (default OFF).

\*LLIST I

Uključuje ili isključuje listanje programa na ekran za vrijeme asembliranja (default OFF).

\*LIST I

Kao \*LIST, ali za štampač.

\*FORM

Šalje „form feed“ znak na printer.

\*TITLE x\$[,x]

Služ za ispis naslova na papiru printera za vrijeme asembliranja (x je željeni broj stranice).

\*MACLIST I

Uključuje ili isključuje prikazivanje koda koji se generiše pomoću makroinstrukcija (default OFF).

\*INCLUDE x\$

Vrši asembliranje sa trake/diska/mikrodravja; uvrštava fajl x\$ na datu poziciju u programu (ima istu ulogu kao \*F u GEN5-u). Kod „spektruma 128“ ova opcija može dati i sa RAM diskom.

\*OPENOUT x\$

Vrši asembliranje na traku/disk/mikrodravj; šalje u kratkim blokovima objektni kod prilikom asembliranja na izlaznu jedinicu. Pomoću INCLUDE/OPENOUT kombinacije uz pomoć mikrodravja ili diska vrlo se jednostavno mogu asemblirati veoma dugački programi.

\*CLOSEOUT

Prekida slanje objektnog koda na izlaznu jedinicu.

\*PROMPTS I

Uključuje ili isključuje ispis „prompt“ poruka (npr. „start tape“) za vrijeme asembliranja (default ON) (Samo za „spektrum“).

\*COUNT I

Uključuje ili isključuje „brojač“ linija za vrijeme asembliranja (default OFF).

\*REPORT I

Uključuje ili isključuje ispis poruka za greške za vrijeme asembliranja (default ON). \*REPORT OFF se redovito koristi kod upotrebe naredbe „Toolkita“ ASSEMBL.

\*CODE I

Uključuje ili isključuje generisanje objektnog koda (default ON).

\*PRINT x\$

Ispisuje string x\$ za vrijeme asembliranja.

\*PAUSE

Pri nailasku na ovu pseudokomandu asembler čeka na pritisak tipke.

\*WHILE x

Početak asemblerske WHILE petlje. Instrukcije između \*WHILE i \*ENDW će se asemblirati sve dok je uslov x ispunjen.

\*ENDW

Kraj WHILE petlje.

\*REPEAT

Početak UNTIL petlje.

\*UNTIL x

Kraj UNTIL petlje. Instrukcije između \*REPEAT i \*UNTIL će se asemblirati sve dok se uslov x ne ispunji.

Pseudoopcije asemblera utiču samo na tok asembliranja i nemaju nikakav uticaj na objektni kod. Stavljaju se unutar asemblerskog listinga, i slično kao kod GEN5-a, počinju zvjezdicom. Slovo i označava fleg. tj. riječ ON ili OFF.

15% veća od verzije u „čistom mašinsku“.

Feniks posjeduje 4 tipa varijabli i nizova: INT, CHAR, PINT i PCHAR. Ovo P je skraćena od POINTER. Varijable i nazivi deklariraju se pomoću naredbe #DS (inače, sve feniks naredbe počinju znakom „#“, odati i naziv „hash extension“). Ova naredba se koristi u obliku:

labela: #DS tip, x

Npr. naredba num: #DS INT, 15 deklarira petnaestocifni cjelobrojni niz „num“. Njegovi članovi su num [1], num[2], itd. Naravno, indeksi u uglastim zagradama mogu biti i izrazi. Navedena broj: #DS INT,1 definiše običnu cjelobrojni promjenljivu „broj“. Naredba slova: #DS CHAR, 40 deklarira string sa imenom „slova“ dužine 40 karaktera. Stringu ne možemo pristupiti u cjelini, nego samo pojedinačnim karakteristikama (npr. string[5]). Na sličan način deklariraju se i pointerске varijable.

Vrlo slična je i naredba #DI. Ona deklarira varijablu i ujedno joj dodjeljuje vrijednost. Možemo, recimo, da stavimo xor: #DI INT, 255, što je, u neku ruku, ekvivalentno sa LET xor:=255. Moguće su i naredbe oblika: #DI CHAR, „računari“. Zanimljivo je da stringovi mogu sadržavati i kontrolne kodove (koj se moraju označiti znakom „\“). Npr. tekst: #DI CHAR, „prvi tekst 13 drugi tekst“.

## Kompajliranje aritmetičkih izraza

Izračunavanje vrijednosti izraza vrši se pomoću naredbi #DSE i #DUE. Prva vrši obracun u predznacnoj, a druga u nepredznacnoj notaciji. Pored već pomenutih operatera izrazi mogu da sadrže i neke nove. Najvažniji od njih je operator „-“ koji služi za dodjeljivanje vrijednosti varijabli (ne treba ga pomešati sa „?“). Ovo je jedini operator koji se obračunava zdesna na lijevo i ima najniži prioritet (1). Naredba #DUE x=y\*y+5 jasna je sama po sebi i odgovara naredbi LET x=y\*y+5. Izrazi se obračunavaju na sličan način kao u C-u, pa su moguće i ovakve kombinacije: #DUE x=y=0 (što je ekvivalentno sa LET x=0: LET y=0 u bejziku), ili čak #DUE x=(y=z\*2)+z-15 (što bi se u bejziku moglo prevesti kao LET y=z\*: LET x=y+z-15). Naredba DUE x=y\*2 z ekvivalentna je bejzиковom LET x=(y\*2).

Naredbe #DSE i #DUE mogu se upotrebiti i bez operatera „=“ (npr. #DUE x+y-1). Tada se rezultata smješta u HL register procesora. Da bismo simulirali DPOKE 30000, x+y iz bejzika, možemo uraditi:

```
#DUE x+y
LD (30000), HL
```

Naredbu, POKE 30000, x možemo simulirati pomoću:

```
#DUE x
LD A,L
LD (30000), A
```

Izrazi mogu da sadrže izmješane varijable sva četiri tipa. Ovaj program

```
broj: #DS INT, 1
string: #DI CHAR, „ab“
#DUE broj=string
[1]+2*string [2]-5
```

ekvivalentan je naredbi LET broj=CODE „a“+2\*CODE „b“-5.

Nova dva operatera pozajmljena iz C-a su „++“ i „--“. Oni uvećavaju ili umanjuju operand za jedan i vraćaju kao rezultat izmjenjenu vrijednost operanda. Npr. #DUE x++ odgovara LET y+=1: LET x=y, a #DUE --x znači LET x=x-1 i nova vrijednost x-y-a se prenosi u HL register.

Unarni operator „&“ označava „ adresu od“. Npr. #DUE ptr=&cnt dodjeljuje varijabli „ptr“ (koja mora biti pointerског типа) adresu varijable „cnt“. Konstrukcija tipa DUE byte->cnt moguća je samo ako je varijabla „count“ pointerског типа. Inače, varijable tipa PINT i PCHAR razlikuju se samo po dejstvu operatera ++ i --. Tip PINT povećava se i smanjuje dejstvom pomenutih operatera za dva (zato što INT varijabla zauzima dva bajta), a ne za jedan.

I još nešto: za prenos vrijednosti iz registra u varijablu moramo se poslužiti sljedećim trikom:

```
work: DEFS 2
var: #DS, INT, 1
LD (work), HL
#DUE var=work
```

Ova sekvenca simulira naredbu LET var=HL.

## Funkcije, uslovi i petlje

Feniks omogućava i definisanje funkcija. One se deklariraju pomoću naredbe #FNC u obliku labela: #FNC tip. Parametri funkcije deklariraju se pomoću naredbe #PRM. Početak definicije funkcije označava se sa #BEGIN, a kraj sa #END. Krajnji rezultat funkcije je ono što se zatekne u HL registru (ili registru L, ako je funkcija bila tipa CHAR). Jedan primjer:

```
square: #FNC INT
x: PRM INT
```

```
y: #PRM INT
#BEGIN
#DUE x*x*y
#END
```

Funkciju možemo pozvati pomoću izraza #DUE z-square (2,3) ili #DUE square (10,z+3)-9 itd. Ispred naredbe #BEGIN mogu se definisati lokalne promjenjive (pomoću #DS). Parametri funkcije su takođe lokalne prirode, tako da se funkcije mogu pozivati rekurzivno. Ukoliko funkcija nema parametara, moraju se navesti prazne zagrade, npr. #DUE t=vrijeme (). Funkcija ne mora da vrati nikakav rezultat. Vrijednost HL registra u tom slučaju ne može se predvidjeti.

Naredbe #IF x, #ELSE i #ENDIF trebale bi da budu jasne i bez objašnjenja. Na sljedećem primjeru sve se lijepo vidi:



## Upozorenja za kraj

Ostale su još naredbe #LIB i #STACK [x]. Naredba #LIB obavezno se mora naći na kraju programa koji je koristio feniks instrukcije. Ona postavlja u objeکتni kod rutine za izvršenje korištenih naredbi feniksa, čime čini objeکتni kod potpuno neovisnim od kompajlera. Izostavite li ovu naredbu, dobićete neku vrstu p-koda i svaki pokušaj startanja programa završiće se fatalno. Naredba #STACK priprema aritmetički stek za korištenje i obično se stavlja na početak programa. Ova naredba nije neophodna u svim slučajevima, ali je vrlo poželjna. Ona generiše sljedeći kod:

```
LD IX,0
ADD IX, SP
DEC SP
```

Ukoliko se zada i parametar x, generišaće se još i LD SP, x instrukcija, samo tada treba biti vrlo oprezan pri povratku u bejziku. Ako je korištena #STACK instrukcija bez parametara, povratku u bejziku treba ostvariti pomoću INC SP i RET.

Još treba obratiti pažnju na jednu vrlo čestu grešku. Naime, ako imamo program:

```
ORG 25500
PUT $
```

```
VAR: #DI INT, 100
start: #DUE var
LD (23728), HL
RET
```

niukom slučaju ga ne treba startati od adrese 25500, nego od vrijednosti labela „start“ (ovdje 25602 što možemo saznati pomoću PRINT start). Dakle, deklaracije varijabli i funkcija treba preskočiti! Najbolje je program pozvati sa EXECUTE start direktno iz editora.

Ovo je, u kratkim crtama, bio pregled assemblera „Laser“. U kratkim crtama??? Da! Originalno uputstvo, ima, sadrži znatno više informacija i dugačak je stolom i kusur stranica posvećenih SAMO ASEMBLERUIII još da kažemo čemu služe programi MPANFCSPHI i TRANS koje dobijete uz assembler. Prvi sadrži definicije funkcija u feniks-u koje omogućavaju rad sa FF brojevima (doduše samo sabiranje, oduzimanje i množenje), a drugi prevodi isječanke blokove stvarne pomoću OPENUT0 I1 SAVE u oblik koji će moći da se učita iz bejzika.

Na kraju, i jedan bag — ne sminjate ništa na mikrodžirvi iz „Lasera“ ukoliko niste sigurni da na kartidžu ima dovoljno mjesta. Ako vam se „Laser“ assembler sviđio, pročitaite i tekst u sljedećem broju „Računara“ posvećen monitoru i analizeru. Iznađenja vas tek očekuju!

```
vec: #FNC INT
x: #PRM INT
```

```
y: #PRM INT
#BEGIN
#IF x>y
#DUE x
#ELSE
#DUE y
#ENDIF
#END
```

Naredba DUE n=veci (5,9) dodjeljuje broj 9 promjenljivoj n.

Takođe, naredbe #WHILE x, #ENDW, #REPEAT i #UNTIL, x znaju već i mala djeca. Sljedeći primjer definiše stepenu funkciju:

```
pow: #FNC INT
x: #PRM INT
y: #PRM INT
local: #DS INT, 1
#BEGIN
#DUE local=1
#REPEAT
#DUE local=local — x
#UNTIL y=0
#DUE local
#end
```

Naredna funkcija ispisuje karakter na ekran, a ne vraća nikakav rezultat:

```
chout: #FNC INT; tip nije važan
broj: #PRM CHAR
#BEGIN
LD A,2
CALL 1601H
#DSE code
LD A, L
RST 16
#END
```

Naredba #RETURN koristi se rjeđe i izaziva bezuslovni povratik iz funkcije bilo gdje unutar BEGIN...END konstrukcije i donekle odgovara naredbi STOP u jeziku LOGO.

Željko Jurčić



# Metodi gradijenta

„Metodima gradijenta“ počinje serija „Računarski algoritmi“ u kojoj će prof. dr Dušan Slavić dati niz algoritama sa programima za rešavanje odabranih numeričkih problema. U ovom tekstu reč je o traženju minimuma (ili maksimuma ili nule ili neke druge vrednosti) funkcije više promenljivih, kao i rešavanju sistema nelinearnih jednačina. Ovaj primer potvrđuje da se i teški numerički problemi mogu pretočiti u jednostavan algoritam i razumljiv program.

Serijski „Računarski algoritmi“ u svakom svom nastavku sadrži obradu nekog numeričkog problema od matematičke formulacije, preko numeričke analize do računarskog algoritma i programa. Nimalo slučajno za početak serije odabran je uvek aktuelan problem nalaženja minimuma (ili neke druge vrednosti) funkcije više promenljivih. To je svima lako razumljiv problem. Ako je broj promenljivih veliki, onda je očigledno neophodno pozvati u pomoć računar, koji je uostalom i izmišljen da bi se rešavali optimizacioni problemi.

Iako je teško dati neku geometrijsku predstavu n-dimenzionalnom „prostoru“, u matematički se itekako mnogo radi sa vektorima X sa n dimenzija. Tako se uvodi vektorski argument X koji uređeni skup od n elemenata  $x_1, \dots, x_n$  uzima vrednosti od 1 po 1 do n ( $i = 1(1)n$ ).

$$X = [x_1 \ x_2 \ \dots \ x_n]^T$$

Skalarne veličine  $x_i$  nazivaju se komponente vektora (ili koordinate ili argumenti). „T“ u formuli znači transponovanje, jer je reč o matrici-koloni kojom se vektor prikazuje. Bitno je jedino da svakoj takvoj uređenoj n-torki brojeva  $x_1, x_2, \dots, x_n$  prema nekom pravilu (koje zovemo „funkcija f“) odgovara neki skalar (koji zovemo „vrednost funkcije f(X)“). Vrednost funkcije f(X) se menja u prostoru i nas zanima za koji argument X funkcija f ima minimum.

Naravno, za potrebe računara moramo u svakom posebnom slučaju zadati računaru namenu funkciju f(X) i od njega tražiti očekivati da nam da rezultat. Kako se još nisu obistinila velika obećanja zagovornika „veštačke inteligencije“ (računari sedme, osme i devete generacije), neophodno je ne samo zadati funkciju f(X), nego i početnu vrednost argumenta X(0), kao i valjan program koji definitivno „nikad“ inteligentnom računaru omogućava da nađe rešenje, iako on nikad neće shvatiti ni šta radi, ni kako radi. Za nas je valjda jedino važno da računari radi ono što mu zadamo da radi i to na način koji mu zadržamo da radi.

Treba naravno znati da li je traženje minimuma jedan od ključnih problema i zašto jeste? Ako umemo da odredimo minimum neke funkcije f, onda nije teško problem nalaženja maksimuma neke funkcije p(X) svesti na prethodni: dovoljno je tražiti minimum funkcije  $f(X) = -p(X)$ . Ako nam je zadat drugi problem: da odredimo X za koje neka funkcija q(X) ima nulu, možemo umesto toga tražiti minimum funkcije  $f(X) = (q(X))^2$ . Ako je potrebno odrediti argument X za koji neka funkcija r(X) ima vrednost s, dovoljno je tražiti minimum

```

10 REM ----- Dušan Slavić, POJEDNOSTAVLJENJE METODA GRADIJENTA
20 REM Zadati N, broj komponenta nepoznatog vektora X
30 N=3 DEFDBL A-H, D-Z DIM X(N), G(N)
40 REM Zadati početnu aproksimaciju X, priraštaj H i broj iteracija M
50 X(1)=2 X(2)=2 X(3)=2 H=2E-19 M=256 I=0 TO 100
60 REM Zadati funkciju f čija se tačka minimuma-nule X traži
70 A=LOG(X(1))-X(2)+X(3) B=X(1)-2*EXP(X(2)-1)+X(3)
  C=SQR(X(1))+X(2)-2*X(3) F=A+A*B*B+C
80 RETURN: REM Kraj zadavanja funkcije f
90 REM Pojednostavljeni metod gradijenta
100 FOR L=1 TO M
110 REM Početak iteracije
120 B0SUB 60: P=F: R=0: FOR I=1 TO N: Q=X(I): X(I)=Q-H: B0SUB 60:
  E=F: X(I)=Q+H: B0SUB 60: G=(F-E)/H: R=R+G*G: G(I)=G: X(I)=Q: NEXT
130 IF R=0 THEN STOP ELSE R=-4*R/I: FOR I=1 TO N: X(I)=B(I)*R+X(I): NEXT
140 REM Kraj iteracije
150 PRINT USING "#####":L;
  FOR I=1 TO N: PRINT USING "#####.#####":X(I); NEXT: PRINT
160 NEXT L: END

```

funkcije  $f(X) = (r(X) - s)^2$ . Zadani sistem nelinearnih jednačina  $Y(X)=0$  može se rešiti i ako se traži minimum funkcije  $f=Y^T Y$ , i tako dalje, mnogi problemi mogu se svesti na traženje minimuma.

Kako naći minimum: kako od neke vrednosti argumenta X(k) koja se u k-toj iteraciji smatra poznatom, preći na narednu vrednost argumenta X(k+1) koja je bliža tački u kojoj funkcija f ima minimum? Ako je broj promenljivih jedan, to nije teško: u nedostatku bolje ideje može se zadati interval podeliti na desetak podintervala, izračunati u graničnim tačkama vrednost funkcije i na osnovu toga zaključiti u kom se podintervalu nalazi tražena tačka. Zatim se taj uočen podinterval dalje deli — dok ne dođemo do zadovoljavajuće tačnosti rezultata. Ako je broj promenljivih n veći od jedan, ovaj metod ne može se promeniti, kao ni oko stotinu drugih poznatih metoda.

Pretraživanja u stilu „metoda Monte-Karlo“ ne pomažu prosto zato što se za veće n nema dovoljno vremena da se za života programera dobije rezultat. Kada se programer nada bespomoćno pred problemom, ponekad se seti onog plenskog vraća koji (očajan što madije ne pomažu) rezignirano zaključuje: ako ne pomogne ni, evo nadiraja f— moraću da primenim penicilin. U našem problemu takode postoje antibiotici, oni se zovu — metodi gradijenta.

U fizici se gradijent koristi, na primer, u prikazivanju električnog polja K koje se uvodi kao gradijent električnog potencijala, sa promenjenim znakom (jer se uzima da polje deluje od tačke na višem potencijalu ka tački na nižem potencijalu).

U matematički se gradijent G obično uvodi definicijom koja komponente  $g_i$  tog

vektora izražava pomoću parcijalnih izvoda funkcije f po argumentima  $x_i$ :

$$G = [g_1 \ g_2 \ \dots \ g_n]^T$$

gde su  $g_i$  parcijalni izvodi

$$g_i = \partial f / \partial x_i, i=1(1)n.$$

Izvod skalarne funkcije u nekom pravcu jednak je projekciji gradijenta na taj pravac. Intenzitet gradijenta g jednak je izvodu skalarne funkcije u pravcu njene najbrže promene. Ako se traži minimum funkcije treba ići u pravcu gradijenta, ali u suprotnom smeru. Zato se metodi gradijenta i nazivaju metodima najstrijelje spusta.

Zasad je jasno da se od stare tačke n-dimenzionalnog prostora X(k) treba krenuti ka novoj tački X(k+1) u pravcu gradijenta određenog u staroj tački G(k).

$$X(k+1) = X(k) - t(k)G(k),$$

ili kako se to u novije vreme (da ne kažem: po poslednjim modji) piše

$$X := X - t \cdot G.$$

Faktor t je ono po čemu se međusobno razlikuju metodi gradijenta. Faktor t ne sme biti negativan, jer bi se inače udaljavalo od tražene tačke. Faktor t ne sme biti suviše veliki, jer bi mogli preskočiti oblast u kojoj se nalazi minimum funkcije. Faktor t ne sme biti suviše mali, jer bi to značilo da moramo činiti ogroman broj koraka — a u svakom treću računati gradijent. Faktor k ne sme biti neki fiksan broj, jer bi za početne iteracije svaki broj bio premalen, a za kasnije prevelik. Kako smo raspoloženi da u doglednoj budućnosti dobijemo rezultat, korak bi svakako trebalo smanjivati na neki sistematičan način. Pojedini autori tekstova iz numeričke analize izbegavaju da daju odgovor na pitanje: kako menjati korak.

U literaturi se može sresti i sledeće određivanje faktora t

$$t = \sum_{i=1}^n \sum_{j=1}^n (f(x_i, y_j))^2 / \sum_{i=1}^n \sum_{j=1}^n (f(x_i, y_j))$$

Ako je posmatrana skalarna funkcija f kvadrat neke vektorske funkcije Y, definisane sa

$$Y = [y_1, y_2, \dots, y_n]^T$$

$$f = Y^T Y = Y^2$$

onda se obično uvodi kvadratna matrica A dimenzije n x n čiji su elementi a<sub>ij</sub>

$$a_{ij} = \partial y_i / \partial x_j$$

i tada se dokazuje da novu tačku treba računati pomoću stare na sledeći način

$$t = (Y^T(AA^T Y)) / (2(AA^T Y)^T(AA^T Y))$$

$$G = 2A^T Y, X = X - t G$$

Za n = 1 ova formula se svodi na Newtonov tangentni metod

$$X = x - y/(dy/dx)$$

Navedeni način određivanja koraka je komplikovan, pa se opredeljujemo za sledeći, jednostavniji

$$G = 2A^T Y, t = Y^T Y / (2Y^T A^T Y)$$

$$X = X - t G$$

I ova formula se, za n = 1, svodi na Newtonov tangentni metod. Ona se može napisati na prostiji način

$$G = \text{grad } f, t = 2f / (G^T G), X = X - t G$$

Kako u poslednjim formulama ne figurise matrica A, ovaj metod je opštiji od ubičajenog, jer se odnosi na proizvoljnu skalarnu funkciju f, a ne samo zbir kvadrata levih strana sistema nelinearnih jednačina YY. Dakle, iteracija glasi

$$G = \text{grad } f, X = X - (2f/G^T) G$$

Ostaje da se odredi način izračunavanja parcijalnih izvoda. Jedan od svakako mogućih načina je da se parcijalni izvodi odrede analitički, pa da se na računaru u svakoj tački izračunavaju vrednosti tih analitički određenih izvoda. To je više posla u pripremi programa (posebno ako se izvode složene funkcije).

Drugi način se više primenjuje: zada se samo funkcija f, a vrednosti parcijalnih izvoda se izračunavaju numerički — manje posle u pripremi ali veća greška u izračunavanju parcijalnih izvoda. Neka su E(i) jedinični vektori duž osa n-dimenzionalnog koordinatnog sistema ili kolone jedinične matrice

$$I = [E(1) E(2) \dots E(n)]$$

Tada se g, obično računajući pomoću prostije formule

$$g_i = (f(X+h^i E(i)) - f(X))/h_i$$

iaiko je tačnija sledeća formula

$$g_i = (f(X+h^i E(i)) - f(X-h^i E(i))) / (2 \cdot h^i)$$

U primeru iz priloženog programa koreni nelinearnih jednačina imaju očekivane vrednosti oko jedinice pa se u poslednjoj formuli uzima da je vrednost priratažeta h oko 10<sup>-8</sup>. Ceo račun obavlja se u dvostrukoj tačnosti. Cilj je naći rešenje sistema jednačina

$$\log(x) - y + z = 0,$$

$$x - 2 \exp(y-1) + z = 0,$$

$$\text{sq}(x) + y - 2z = 0.$$

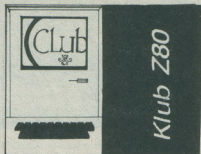
ili, što je isto, minimum funkcije

$$f = (\log(x) - y + z)^2 + (x - 2 \exp(y-1) + z)^2 + (\text{sq}(x) + y - 2z)^2.$$

Početna aproksimacija vektora X je: x=2, y=2, z=2. Rešenje sistema jednačina je x=1, y=1, z=1.

Dušan Yavić

55/računarski algoritmi



## Kako štampati ime

Ideja se začela kad je u redakciju stiglo pismo nepotpisanog vlasnika spektruma iz Zemuna.

„Prilož sam brdo priručnika i članaka o mašinskom jeziku... arhitekturi Z80, alternativnim registrima, logičkim operacijama, bitovima, binarnim brojevima... Imam četiri kompleta mašinskih naredbi, jedna od njih je iz „Računara“. Jasno mi je šta znači LD A, (HL), ali ne umem na mašincuu da napišem ni „PRINT moje ime i prezime“. Nema nam ništa protiv sistemskog softvera i genijalnih rutina za super-programere, ali zar „Računari“ zaista ne mogu malo prestora da posvete i nama početnicima?“

„Kad gledam popis mašinskih naredbi kao da mi je neko u ruke dao rečnik

```

10  +-----+
20  +
30  + PRINT 1 +
40  +
50  +-----+
60  ORG 50000
70  LD A,2
80  CALL #1601
90  LD HL,TEXT
100 CALL PRINT
110 RET
-----
120 PRINT LD A,(HL)
140 CP #8
150 RET 2
160 RST 16
170 INC HL
180 JR PRINT
190
200 TEXT DEFB 22,0,0
210 DEFB "PRIMER KAKO"
220 DEFB 22,10,9
230 DEFB "SE PRINTUJE"
240 DEFB 22,21,22
250 DEFB "NEKI TEKST"

```

stranog jezika, a ja sad treba na tom jeziku da progovorim. Ni jedan jezik, pa ni programski, ne uči se tako što će se najpre nabubati sve reči. Bez konverzije one može se napred..."

Pismo je imalo četiri gusto kucane stranice, ali i ovo što smo naveli dovoljno je rečito. Oduka je pala:

1. Osniva se KLUB Z80.
2. Osnovna delatnost Kluba je: konverzija na mašincuu.
3. Članovi kluba mogu biti svi „spektrumovci“ koji nešto (mnogo, malo) znaju ili nešto (mnogo, malo) hoće da nauče o mašinskom jeziku.
4. Sa dužnom pažnjom i zahvalnošću primaju se i prihvataju ili odbacuju svi priloz i obliku:
  - a) Pitanja, želja, predloga, zahteva;
  - b) Listinga koji se mogu objaviti u „Računarima“.

```

10  +-----+
20  + PRINT 2 +
30  +
40  +
50  +-----+
60  ORG 50000
70  LD DE,16384
80  LD IX,TEXT1
90  CALL PRINT
100 LD DE,18505
110 LD IX,TEXT2
120 CALL PRINT
130 LD DE,20662
140 LD IX,TEXT3
150 CALL PRINT
160 RET
-----
170 PRINT LD A,(IX)
190 CP #0
200 RET Z
210
220 LD H,0
230 LD A,A
240 ADD HL,HL
250 ADD HL,HL
260 ADD HL,HL
270 LD BC,(23606)
280 ADD HL,BC
290
300 PUSH DE
310 LD B,8
320 PTL LD A,(HL)
330 LD (DE),A
340 INC D
-----
350 INC HL
360 DJNZ PTL
370 POP DE
380
390 INC DE
400 INC IX
410 JE PRINT
420
430 TEXT1 DEFB "PRIMER KAKO"
440 TEXT2 DEFB "SE PRINTUJE"
450 TEXT3 DEFB "NEKI TEKST"

```

Priloz pod b) se honorišu, ukoliko je uz listing poslata i kaseta sa snimljenim programom i, naravno, ukoliko se objave.

5. Ni jedan programski problem ne može biti banalan ili kompleksan do te mere da ne zaslužuje razmatranje u KLUBU Z80.

Kao odgovor nepotpisanom Zemuncu, Klub u ovom broju objavljuje dve rutine pomoću kojih može na mašincuu da ispiše svoje-ime-i-prezime.

Rutina PRINT 1 je kraća i jednostavnija. Za ispis se koristi spektrumov čuveni restart 16, odnosno potprogram iz ROM-a koji se zove PRINT-A-1.

Nešto složenija i duža, ali zato mnogo brža jeste rutina PRINT 2. Zato što ne poziva ROM, osim što ispituje jednu sistemsku promenljivu, ova rutina izbegava razne provere i izračunavanja. Bajtovе video memorije puni neposredno, pa nije potrebno otvaranje kanala, a nesmetani print je moguć na sva 24 reda. Pošto ne ispituje ni da li je kôd karaktera u dozvoljenom opsegu (32-127), mogućno je formirati svoj set karaktera od 256 znakova (s kodovima 0-255). No, ništa nije samo dobro ili samo loše. PRINT 2 zahteva potpuno poznavanje organizacije ekrana. Pošto nema nikakve kontrole, svaka greška programera može biti fatalna po program. Prelazak ispisa iz jedne trećine ekrana u drugu nije moguć — az neposredne intervencije programera.

Naravno, ovo ni izdaleka nije sve što bi se moglo reći o ispisu i radu sa video memorijom. U obe rutine se, na primer, podrazumeva da su INK i PAPER navedeni definisani. Ali, u činjenici da uvek ima mnogo da se ke kaže i da se sve može reći na više načina leži šansa da Klub Z80, uz svesrdnu pomoć čitalaca, istinski zaživi.

Pripremljuju:  
Aleksandar Radovanović  
Žarko Vukosavljević

# osam krugova kredom

Kružnica je u potpunosti određena poluprečnikom i koordinatama centra, a to se vidi i iz sledeće definicije:

Kružnica je skup tačaka u ravni koje su podjednako udaljene od jedne tačke — centra, a to rastojanje je određeno poluprečnikom kružnice.

## Kako nacrtati krug?

Crtač će uzeti svoj šestar u ruke i primerom nam odgovoriti na pitanje. Na računaru ovo neće ići baš tako lako, pogotovo što proizvođači još uvek ne isporučuju pribor za crtanje u osnovnim konfiguracijama. Šalu na stranu, ovo zalista nije jednostavan problem, ukoliko se ne shvati olako. Ipak, postoji nekoliko osobina koje će svaki, na računaru nacrtati, krug posedovati. Otkrpio je da će neke tačke biti bliže, a neke opet dalje od centra. Krug se može nacrtati onoliko precizno koliko to dozvoljavaju tehničke mogućnosti računara, odnosno rezolucija grafičkog ekrana. Sasvim je normalno zahtevati da se kružnice iscrtavaju baš toliko precizno. Zbog toga je potrebno svaku tačku posebno crtati, jer bilo kakve dodatne aproksimacije u vidu spajanja tačaka linijama neizbežno dovode do loše iscrtane krive. Ovo ne važi samo u jednom posebnom slučaju, koji će kasnije biti razmatran.

Kos smerni crtanje neke bitnije razlike ne mogu postojati, tj. na tom polju nije moguće izmisliti nešto novo što bi značajnije ubrzalo posao. Zato ćemo se orijentisati tačaka kruga. Zaista postoji mnogo načina da se izvrše proračuni, ali, treba se pobrinuti da bude izabran najbolji.

Razvoj programa je prikazan u osam stapa, a svaka od njih ima oblik procedure napisane na paskalu (slika 1). I pored nesumnjivo veće popularnosti bejzika, autor ovog teksta se odlučio za paskal, jer je u pitanju prenosiviji jezik. Za pojedine računare potrebno je prilagoditi samo hardverski orijentisane procedure;

**plot (x; y; Integer)** — osvetljava tačku čije su koordinate x i y

**move (x; Integer)** — pomera grafički kursor na poziciju x,y

**line (x; Integer)** — povlači liniju između grafičkog kursora i tačke x,y

**origin (x; Integer)** — postavlja centar koordinatnog sistema na poziciju x,y (Napomena: sve operacije se obavljaju relativno od trenutnog centra koordinatnog sistema — ishodišta — koji se postavlja procedurom **origin**.)

Sve ostalo će na svim računarsima raditi podjednako dobro. Da se ne bismo zadržavali na beznačajnim detaljima, smatraćemo da je centar kruga istovremeno i centar koordinatnog sistema. Ovo će nam omogućiti da se potpuno posvetimo problemu koji nas zanima.

Najzad, krenimo sa prikazom pojedinih algoritama, naravno, od najlošijeg ka najboljem.

## Trigonometrija

Odmah se nameće, a da niko ne zna tačno zbog čega, korišćenje trigonometrijskih funkcija — sinusa i kosinusa. Proceduru **Circle**, tačnije — njen ekvivalent pisan u bejziku, verovatno ćete naći u svakom priručniku koji se dobija uz kupljenu računaru. Karakteristično je za sve algoritme ove vrste da je kontrolna promenljiva petlje ugao (u stvari, radi se o centralnom uglu). Pošto je rešeno da se kružnice crtaju maksimalno precizno, potrebno je odrediti koliko tačaka će sadržati kružnica ili, bolje rečeno, koliki će biti korak kojim se menja ugao. Korak ne sme biti preveliki, jer će kružnica biti isprekidana, a ako bude isuvi-

Tabela vremena

Circle1	17.05
Circle2	5.04
Circle3	3.05
Circle4	1.57
Circle5	1.65
Circle6	1.65
Circle7	0.97
Circle8	0.78
Circle9	0.65
Fast plot	0.21

še mali, crtanje će se nepotrebno odužiti. Tražena vrednost se može dobiti primenom formule za računanje obima kruga

$$O = 2 \cdot \pi \cdot r, \text{ odakle je } \\ \text{anglestep} = 2 \cdot \pi / O \cdot 1/r$$

i takvim načinom ćemo se zadovoljiti. Sada se promenljiva **angle** kreće od 0 do  $2 \cdot \pi$  sa korakom **anglestep**. Sve radi savršeno, ali nevolja je u tome što na ovom svetu ima sve više nervoznih i nestrijpljivih ljudi. Kamo sreće da imamo „Kreja“, ili nekog sličnog diva, jer u tom slučaju ne bismo morali dalje da se brinemo za sudbinu ovog problema.

Prvo, a odmah zatim i drugo, poboljšanje je čisto tehničke prirode, a sastoji se u smanjenom obimu računanja. Zaista, sasvim je dovoljno računati trigonometrijske funkcije samo tako gde centralni ugao šeta interval od 0 do  $\pi/4$ , sve ostale vrednosti se jednostavno dobijaju promenama znaka i razmenom vrednosti apscise i ordinate. Tako se računaju jedan par (x,y) vrednosti i crta osam tačaka {x; x;-y; -x;-y; -x;-y; y;-x; -y;-x}. Na ovaj način vreme potrebno za izračunavanje smanjeno je osam puta. Ovo se može potkrepiti i osnovnim trigonometrijskim jednačinama koje pristižu iz osobina ove dve funkcije:

$$\cos(x) = -\cos(\pi - x) = -\cos(\pi + x) = \cos(-x) \\ \sin(x) = \sin(\pi - x) = -\sin(\pi + x) = -\sin(-x)$$

```

10  org #500
20  ld  bc,ax
30  ld  bl,ax
40  jr  loop
50  defw 0
60  defw 0
70  jr  circle
80  jr  ellipse
90  jr  circle
100 jr  ellipse
110 name: defw "CIRCLE"
120 defw "E"
130 defw "ELLIPSE"
140 defw "E"
150 defw "CIRCLE"
160 defw "E"
170 defw "ELLIPSE"
180 defw "E"
190 defw 0
200
210 main: ld  iy,flag
220
230 call pwp,pa  ;postavlja broj parametara
240 call pwp,op  ;postavlja put
250 call pwp,or  ;postavlja novi oridiz
260 call pwp,io  ;postavlja de iz su ope
                ;postavlja
270 jr  sx,anglep
280 call get,ra  ;iskupiti poluprečnike
290 ld  iy,w
300 ld  iofrast,bl
310 ld  bc,w  ;correct = 0
320 ld  bl,w
330 jr  astay
340 loop: add  bl,de
350 push bc
360 call plot,ta  ;crtati tačku
370 pop bc
380 call next_coord ;dodaj sledeće koordinate
390 entry: add  a,w
400 sw  de,bl
410 shw bl,de
420 jr  p,loop
430 end:ld  de,oldordr ;whisk y=x
440 ld  bl,oldordr
450 call astay
460 ld  de,(cos,w) ;postavlja startni oridiz.
470 ld  de,(cos,y)
480 call astayp  ;nova pozicija
490 bit  w,ty
500 rsh a,(oldang)
510 ld  a,(oldang)
520 jr  astayp  ;i azuda pes
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

$$\sin(x) = \cos(\pi/2 - x) \\ \cos(x) = \sin(\pi/2 + x)$$

Ma kako trivijalno bilo, ovo poboljšanje je izuzetno značajno, i to ne samo zbog



Devet krugova pakia, okrugli sto kralja Artura i hulahop, samo dokazuju koliki se značaj pridaje jednom geometrijskom ilku — krugu. Još od starih Grka, nije bilo matematičara koji se nije bavljavju njime i proučavao mu svojstve. Ni programeri nisu mogli da ostanu po strani, pa je razvijen čitav niz algoritama na njihovo crtanje na računaru. Ovim tekstom ćemo pokušati da prikažemo evoluciju algoritma za crtanje kružnice i da upozorimo na to koliko su ideje i znanje superiorni u odnosu na zanatski pristup programiranju.

```

1000 cv_x=0
1000 c_y=0: b1=1: (x=1)
1001 r=0
1002 i=0
1003 c_r=1: id w: (x=1)
1004 r=0
1005
1006 cv_x=0
1006 cv_y=0: id 1: (x=0)
1007 r=0
1008
1009 cv_x=0
1009 cv_y=0: id 1: (x=0)
1010 r=0
1011
1012 cv_x=0
1012 cv_y=0: id 1: (x=0)
1013 r=0
1014
1015 cv_x=0
1015 cv_y=0: id 1: (x=0)
1016 r=0
1017
1018 cv_x=0
1018 cv_y=0: id 1: (x=0)
1019 r=0
1020
1021 cv_x=0
1021 cv_y=0: id 1: (x=0)
1022 r=0
1023
1024 cv_x=0
1024 cv_y=0: id 1: (x=0)
1025 r=0
1026
1027 cv_x=0
1027 cv_y=0: id 1: (x=0)
1028 r=0
1029
1030 cv_x=0
1030 cv_y=0: id 1: (x=0)
1031 r=0
1032
1033 cv_x=0
1033 cv_y=0: id 1: (x=0)
1034 r=0
1035
1036 cv_x=0
1036 cv_y=0: id 1: (x=0)
1037 r=0
1038
1039 cv_x=0
1039 cv_y=0: id 1: (x=0)
1040 r=0
1041
1042 cv_x=0
1042 cv_y=0: id 1: (x=0)
1043 r=0
1044
1045 cv_x=0
1045 cv_y=0: id 1: (x=0)
1046 r=0
1047
1048 cv_x=0
1048 cv_y=0: id 1: (x=0)
1049 r=0
1050
1051 cv_x=0
1051 cv_y=0: id 1: (x=0)
1052 r=0
1053
1054 cv_x=0
1054 cv_y=0: id 1: (x=0)
1055 r=0
1056
1057 cv_x=0
1057 cv_y=0: id 1: (x=0)
1058 r=0
1059
1060 cv_x=0
1060 cv_y=0: id 1: (x=0)
1061 r=0
1062
1063 cv_x=0
1063 cv_y=0: id 1: (x=0)
1064 r=0
1065
1066 cv_x=0
1066 cv_y=0: id 1: (x=0)
1067 r=0
1068
1069 cv_x=0
1069 cv_y=0: id 1: (x=0)
1070 r=0
1071
1072 cv_x=0
1072 cv_y=0: id 1: (x=0)
1073 r=0
1074
1075 cv_x=0
1075 cv_y=0: id 1: (x=0)
1076 r=0
1077
1078 cv_x=0
1078 cv_y=0: id 1: (x=0)
1079 r=0
1080
1081 cv_x=0
1081 cv_y=0: id 1: (x=0)
1082 r=0
1083
1084 cv_x=0
1084 cv_y=0: id 1: (x=0)
1085 r=0
1086
1087 cv_x=0
1087 cv_y=0: id 1: (x=0)
1088 r=0
1089
1090 cv_x=0
1090 cv_y=0: id 1: (x=0)
1091 r=0
1092
1093 cv_x=0
1093 cv_y=0: id 1: (x=0)
1094 r=0
1095
1096 cv_x=0
1096 cv_y=0: id 1: (x=0)
1097 r=0
1098
1099 cv_x=0
1099 cv_y=0: id 1: (x=0)
1100 r=0

```

```

2100 cv_x=0
2100 cv_y=0: id 1: (x=0)
2101 r=0
2102
2103 cv_x=0
2103 cv_y=0: id 1: (x=0)
2104 r=0
2105
2106 cv_x=0
2106 cv_y=0: id 1: (x=0)
2107 r=0
2108
2109 cv_x=0
2109 cv_y=0: id 1: (x=0)
2110 r=0
2111
2112 cv_x=0
2112 cv_y=0: id 1: (x=0)
2113 r=0
2114
2115 cv_x=0
2115 cv_y=0: id 1: (x=0)
2116 r=0
2117
2118 cv_x=0
2118 cv_y=0: id 1: (x=0)
2119 r=0
2120
2121 cv_x=0
2121 cv_y=0: id 1: (x=0)
2122 r=0
2123
2124 cv_x=0
2124 cv_y=0: id 1: (x=0)
2125 r=0
2126
2127 cv_x=0
2127 cv_y=0: id 1: (x=0)
2128 r=0
2129
2130 cv_x=0
2130 cv_y=0: id 1: (x=0)
2131 r=0
2132
2133 cv_x=0
2133 cv_y=0: id 1: (x=0)
2134 r=0
2135
2136 cv_x=0
2136 cv_y=0: id 1: (x=0)
2137 r=0
2138
2139 cv_x=0
2139 cv_y=0: id 1: (x=0)
2140 r=0
2141
2142 cv_x=0
2142 cv_y=0: id 1: (x=0)
2143 r=0
2144
2145 cv_x=0
2145 cv_y=0: id 1: (x=0)
2146 r=0
2147
2148 cv_x=0
2148 cv_y=0: id 1: (x=0)
2149 r=0
2150
2151 cv_x=0
2151 cv_y=0: id 1: (x=0)
2152 r=0
2153
2154 cv_x=0
2154 cv_y=0: id 1: (x=0)
2155 r=0
2156
2157 cv_x=0
2157 cv_y=0: id 1: (x=0)
2158 r=0
2159
2160 cv_x=0
2160 cv_y=0: id 1: (x=0)
2161 r=0
2162
2163 cv_x=0
2163 cv_y=0: id 1: (x=0)
2164 r=0
2165
2166 cv_x=0
2166 cv_y=0: id 1: (x=0)
2167 r=0
2168
2169 cv_x=0
2169 cv_y=0: id 1: (x=0)
2170 r=0
2171
2172 cv_x=0
2172 cv_y=0: id 1: (x=0)
2173 r=0
2174
2175 cv_x=0
2175 cv_y=0: id 1: (x=0)
2176 r=0
2177
2178 cv_x=0
2178 cv_y=0: id 1: (x=0)
2179 r=0
2180
2181 cv_x=0
2181 cv_y=0: id 1: (x=0)
2182 r=0
2183
2184 cv_x=0
2184 cv_y=0: id 1: (x=0)
2185 r=0
2186
2187 cv_x=0
2187 cv_y=0: id 1: (x=0)
2188 r=0
2189
2190 cv_x=0
2190 cv_y=0: id 1: (x=0)
2191 r=0
2192
2193 cv_x=0
2193 cv_y=0: id 1: (x=0)
2194 r=0
2195
2196 cv_x=0
2196 cv_y=0: id 1: (x=0)
2197 r=0
2198
2199 cv_x=0
2199 cv_y=0: id 1: (x=0)
2200 r=0

```

```

3100 cv_x=0
3100 cv_y=0: id 1: (x=0)
3101 r=0
3102
3103 cv_x=0
3103 cv_y=0: id 1: (x=0)
3104 r=0
3105
3106 cv_x=0
3106 cv_y=0: id 1: (x=0)
3107 r=0
3108
3109 cv_x=0
3109 cv_y=0: id 1: (x=0)
3110 r=0
3111
3112 cv_x=0
3112 cv_y=0: id 1: (x=0)
3113 r=0
3114
3115 cv_x=0
3115 cv_y=0: id 1: (x=0)
3116 r=0
3117
3118 cv_x=0
3118 cv_y=0: id 1: (x=0)
3119 r=0
3120
3121 cv_x=0
3121 cv_y=0: id 1: (x=0)
3122 r=0
3123
3124 cv_x=0
3124 cv_y=0: id 1: (x=0)
3125 r=0
3126
3127 cv_x=0
3127 cv_y=0: id 1: (x=0)
3128 r=0
3129
3130 cv_x=0
3130 cv_y=0: id 1: (x=0)
3131 r=0
3132
3133 cv_x=0
3133 cv_y=0: id 1: (x=0)
3134 r=0
3135
3136 cv_x=0
3136 cv_y=0: id 1: (x=0)
3137 r=0
3138
3139 cv_x=0
3139 cv_y=0: id 1: (x=0)
3140 r=0
3141
3142 cv_x=0
3142 cv_y=0: id 1: (x=0)
3143 r=0
3144
3145 cv_x=0
3145 cv_y=0: id 1: (x=0)
3146 r=0
3147
3148 cv_x=0
3148 cv_y=0: id 1: (x=0)
3149 r=0
3150
3151 cv_x=0
3151 cv_y=0: id 1: (x=0)
3152 r=0
3153
3154 cv_x=0
3154 cv_y=0: id 1: (x=0)
3155 r=0
3156
3157 cv_x=0
3157 cv_y=0: id 1: (x=0)
3158 r=0
3159
3160 cv_x=0
3160 cv_y=0: id 1: (x=0)
3161 r=0
3162
3163 cv_x=0
3163 cv_y=0: id 1: (x=0)
3164 r=0
3165
3166 cv_x=0
3166 cv_y=0: id 1: (x=0)
3167 r=0
3168
3169 cv_x=0
3169 cv_y=0: id 1: (x=0)
3170 r=0
3171
3172 cv_x=0
3172 cv_y=0: id 1: (x=0)
3173 r=0
3174
3175 cv_x=0
3175 cv_y=0: id 1: (x=0)
3176 r=0
3177
3178 cv_x=0
3178 cv_y=0: id 1: (x=0)
3179 r=0
3180
3181 cv_x=0
3181 cv_y=0: id 1: (x=0)
3182 r=0
3183
3184 cv_x=0
3184 cv_y=0: id 1: (x=0)
3185 r=0
3186
3187 cv_x=0
3187 cv_y=0: id 1: (x=0)
3188 r=0
3189
3190 cv_x=0
3190 cv_y=0: id 1: (x=0)
3191 r=0
3192
3193 cv_x=0
3193 cv_y=0: id 1: (x=0)
3194 r=0
3195
3196 cv_x=0
3196 cv_y=0: id 1: (x=0)
3197 r=0
3198
3199 cv_x=0
3199 cv_y=0: id 1: (x=0)
3200 r=0

```

vidnog ubrzanja, već i zbog uticaja koji će izvršiti na dalji razvoj ovog algoritma, što je mnogo značajnije. Razlike u brzini mogu se uočiti upoređivanjem prve tri procedure:

**Circle1, Circle2 i Circle3.**

Sada ćemo morati da napustimo izuzetno zanimljivo trigonometrijsko polje, jer su tu sve mogućnosti manje-više iscrpljene.

```

10 (a1-c-8)
20 PROCEDURE kruzgovi;
30 CONST P1=3.141593;
40 VAR x,y,r,c: integer;
50
60 PROCEDURE plot(x,y: integer);
70 BEGIN
80   r:=abs(x)+y;
90   user(#bba);
100 END;
110
120 PROCEDURE movetx,y: integer;
130 BEGIN
140   r:=abs(x)+y;
150   user(#bbc);
160 END;
170
180 PROCEDURE linetx,y: integer;
190 BEGIN
200   r:=abs(x)+y;
210   user(#bbf);
220 END;
230
240 PROCEDURE originx,y: integer;
250 BEGIN
260   r:=abs(x)+y;
270   user(#bbc);
280 END;
290
300 PROCEDURE plot0(x,y: integer);
310 BEGIN
320   plot(x,y);plot(x,-y);plot(-x,y);plot(-x,-y);
330   plot(x,1);plot(-x,1);plot(-x,1);plot(-y,-x);
340 END;
350
360 PROCEDURE line0(x,y,ox,oy: integer);
370 BEGIN
380   IF y=oy
390     THEN plot0(x,oy)
400     ELSE BEGIN
410       movetx(ox,y);linetx(y);movetx(-ox,y);linetx(-y);
420       movetx(ox,-oy);linetx(x,y);movetx(-ox,-oy);linetx(-x,-y);
430       movetx(oy,ox);linetx(x);movetx(-ox);linetx(-x);
440       movetx(-oy,ox);linetx(-y,x);movetx(-oy,-ox);linetx(-y,-x);
450     END;
460 END;
470
480 PROCEDURE Circle1(r: integer);
490 VAR angle,angstep: real;
500 BEGIN
510   angle:=0;angstep:=1/r;
520   REPEAT
530     plot(round(r*cos(angle)),round(r*sin(angle)));
540     angle:=angle+angstep;
550   UNTIL angle=2*PI+angstep;
560 END;
570
580 PROCEDURE Circle2(r: integer);
590 VAR angle,angstep: real;
600   x,y: integer;
610 BEGIN
620   angle:=0;angstep:=1/r;
630   REPEAT
640     x:=round(r*cos(angle));y:=round(r*sin(angle));
650     plot(x,y);plot(x,-y);plot(-x,y);plot(-x,-y);
660     angle:=angle+angstep;
670   UNTIL angle=PI/2+angstep;
680 END;
690
700 PROCEDURE Circle3(r: integer);
710 VAR angle,angstep: real;
720   x,y: integer;
730 BEGIN
740   angle:=0;angstep:=1/r;
750   REPEAT
760     x:=round(r*cos(angle));y:=round(r*sin(angle));
770     x:=plot(x,y);
780     angle:=angle+angstep;
790   UNTIL angle=PI/4+angstep;
800 END;
810
820 PROCEDURE Circle4(r: integer);
830 VAR sq: real;
840   x,y: integer;
850 BEGIN
860   y:=0;x:=r;sq:=sqrt(r);
870   WHILE y<=30
880     BEGIN
890       x:=round(sqrt(sq-sqrt(y)));
900       plot(x,y);
910       y:=y+1;
920   END;
930 END;
940
950 PROCEDURE Circle5(r: integer);
960 VAR sq,sqy: real;
970   x,y: integer;
980 BEGIN
990   y:=0;x:=r;sq:=sqrt(r);sqy:=0;
1000  WHILE y<=30
1010    BEGIN
1020     x:=round(sqrt(sq-sqy));
1030     plot(x,y);
1040     sqy:=sqy+2+1;
1050     y:=y+1;
1060   END;
1070 END;
1080
1090 PROCEDURE Circle6(r: integer);
1100 VAR x,y,left,right: integer;
1110   sq: real;
1120 BEGIN
1130   x:=r;y:=0;correct:=0;
1140   WHILE y<=30 DO
1150     BEGIN
1160       plot(x,y);
1170       left:=round(sqrt(r)-sqrt(x-1)-sq);
1180       right:=round(sqrt(y)+sqrt(x)-sq);
1190       IF abs(left|cab|right)
1200         THEN x:=x-1;
1210     END;
1220   END;
1230 END;
1240
1250 PROCEDURE Circle7(r: integer);
1260 VAR x,y,correct,left,right: integer;
1270 BEGIN
1280   x:=r;y:=0;correct:=0;
1290   WHILE y<=30 DO
1300     BEGIN
1310       plot(x,y);
1320       left:=correct+y-x+1+2;
1330       right:=correct+y+2+1;
1340       IF abs(left|cab|right)
1350         THEN BEGIN
1360           correct:=left;
1370           x:=x-1;
1380         END
1390         ELSE correct:=right;
1400       y:=y+1;
1410     END;
1420 END;
1430
1440 PROCEDURE Circle8(r: integer);
1450 VAR x,y,correct,left,right,oldx,oldy: integer;
1460 BEGIN
1470   x:=r;y:=0;oldx:=x;oldy:=y;correct:=0;
1480   WHILE y<=30 DO
1490     BEGIN
1500       right:=correct+y+2+1;
1510       left:=right+2+1;
1520       IF left>right DO
1530         THEN BEGIN
1540           correct:=left;
1550           line(x,y,oldx,oldy);
1560           x:=x-1;
1570           y:=y+1;
1580           oldx:=x;
1590           oldy:=y;
1600         END
1610         ELSE BEGIN
1620           correct:=right;
1630           y:=y+1;
1640         END;
1650   END;
1660   plot(x,y);
1670 END;
1680
1690 BEGIN
1700   REPEAT
1710     write('Radijus: ');read(r);
1720     write('Dot program: ');read(c);
1730     page:=origins(320,200);
1740     CASE c OF
1750       1: Circle1(r);
1760       2: Circle2(r);
1770       3: Circle3(r);
1780       4: Circle4(r);
1790       5: Circle5(r);
1800       6: Circle6(r);
1810       7: Circle7(r);
1820       8: Circle8(r);
1830     UNTIL false;
1840 END.

```

Zato čemo, sa posebnim zadovoljstvom, pozvati u pomoć matematičarke antičke Grčke.

### Pitagorina teorema

Uočimo neke značajne osobine kružnog luka, onog gde je centralni ugao u intervalu  $[0, \pi/4]$ , jer je on sada jedini od interesa za ovu našu raspravu. U tom intervalu vrednosti apcise se kreću od  $r$  do  $r/\sqrt{2}$ , a ordinatne od  $0$  do  $r/\sqrt{2}$ . Pri tom treba zapaziti da se ordinata konstantno menja sa korakom od jedne tačke. Na razmatranje obeležja ovog kružnog luka još jednom čemo se vratiti, a dosadašnji zaključci nam

omogućavaju primenu Pitagorine teoreme. Naravno, potrebno je prethodno ustanoviti u kom se odnosu nalaze apscisa i ordinata, kao udaljenosti tačke na kružnici od  $x$  odnosno  $y$ -ose. Vidi se da je u pitanju pravougli trougao, čija hipotenuza je poluprečnik, a dužine kateta su vrednosti koje imaju apscisa i ordinata u datoj tački. Primerom Pitagorine teoreme dolazimo do sledećih jednačina:

$$\begin{aligned} \text{sqr}(r) &= \text{sqr}(x) + \text{sqr}(y) \\ x &= \text{sqr}(\text{sqr}(r) - \text{sqr}(y)) \end{aligned}$$

(Napomena: promenljiva  $x$  uzima samo pozitivne vrednosti korena. Funkcija  $\text{sqr}$  (square root) je operacija kvadratnog kore-

na, a  $\text{sqr}$  (square) operacija kvadriranja.)

Kvadrat poluprečnika je konstanta, a ordinata (promenljiva  $y$ ) kreće od  $0$  sa konstantnim korakom  $1$ . Na ovaj način, umesto dve trigonometrijske operacije, račun se samo jedan koren i jedan kvadrat. Sve ovo je prikazano u proceduri **Circle4**.

Sledeća procedura (**Circle5**) predstavlja malu digresiju, ali je u svakom slučaju zanimljiva. Radi se o nešto poboljšanoj proceduri **Circle4**, a poboljšanje se sastoji u izbegavanju računanja kvadrata promenljive  $y$ . Posle jedinačke koja sledi, stvari će verovatno biti mnogo jasnije.

$$\text{sqr}(y+1) = \text{sqr}(y) + 2^*y + 1$$

U pitanju je sukcesivno izračunavanje kvadrata prirodnih brojeva. Iz prethodne jednačine se vidi da se to može izvršiti iterativnom metodom, a koristi se, praktično, samo operacija sabiranja. Ako nam je poznat kvadrat prethodnog u nizu prirodnih brojeva, vrlo jednostavno dolazimo do vrednosti koja nam je potrebna.

Kao što se vidi iz tabele u kojoj su prikazana vremena (slika 2), procedura *Circles5* je nešto sporija od svoje prethodne. Uzrok tome je povećan obim adresiranja promenljivih. I pored toga, može se reći da ona predstavlja nesumnjiv napredak u odnosu na proceduru *Circles4*.

Ovaj vrlo specifičan slučaj izračunavanja kvadrata pokazao se i kasnije kao koristan, mada u ponešto izmenjenom obliku, i na neki čudan, neočekivan način usmerio ova istraživanja ka njihovom krajnjem cilju.

## Malo logike

Osvrnimo se na ono što smo do sada postigli. U prvoj proceduri za svaku tačku izračunavaju se dve trigonometrijske funkcije, a u poslednjoj na svakih osam tačaka jedan koren i nešto množenja i sabiranja. Ipak, može se uočiti i jedna zakonitost — u svakoj proceduri se pojavljuje funkcija *round*. To znači da smo svuda imali po neki broj koji pred celog ima i razlomljeni deo. Ovo je svakako najbolja osobina dosadašnjih procedura. Sledeći potez moramo tako usmeriti da se najzad oslobodimo realnih brojeva i da se nađemo na polju celih brojeva, jer se njima manipuliše na mnogo prirodniji i brži način.

Vratimo se na ono značajno osmini kruga i pokušajmo da uočimo osobine koje će nam pomoći u našim namerama. Promenljive  $x$ , odnosno ordinata, menja se, kao što je već utvrđeno, na jednostavan način, konstantnim korakom veličine jedne tačke. Sa druge strane promenljiva  $y$  (apscisa), menja se u zavisnosti od promenljive  $x$  ali nešto posmatranom intervalu. To se može postaviti i drugačije: da li se promenljiva  $x$  smanji za jedan ili oстане nepromenjena. U poslednjem rečenici nazire se slab i pomalo nejasan znak binarne logike, ali ipak dovoljan da nam osvetli put i posledica toga je procedura *Circles6*.

Analizirajmo je. Posle deklariranja promenljivih, inicijalizuje se:  $x$  na  $r$  (radijus) i  $y$  na 0. Kontrolna struktura je *WHILE DO*, a uslov izražava  $y > x$ . Pošto već imamo neke korisne koordinate (one koje pripadaju krugu), nema nikakvih prepreka da se na crtaju potrebne tačke. Sada je potrebno odrediti sledeće vrednosti promenljivih  $x$  i  $y$ . Ovaj problem se svodi na jednu dilemu: da li se potrebno smanjiti promenljivu  $x$ , ili je ostaviti nepromenjen. Prava vrednost promenljive  $x$  biće  $\text{sq}(r-x)$  (gdje  $\text{sq}(x) = x^2$ ), međutim, ona nam nije potrebna. Sasvim je dovoljan kao deo, a do njega ćemo doći na sledeći način. Neka je  $x$  prava vrednost koja zadovoljava sledeću jednačinu:

$$\text{sq}(r) = \text{sq}(x) + \text{sq}(y+1)$$

Ali, vrednost  $x$  ćemo morati da zamenimo (aproksimiramo) celim brojem, a u kombinaciji su jedino brojevi  $x$  i  $x-1$ . Potrebno je odlučiti se za jednog od njih, ili, drugim rečima, smanjiti promenljivu  $x$  ili ne. Vrlo jednostavno, uzemoćemo onu vrednost koja je bolje uklapa, tj. koja je približnija vrednosti  $x$ . Vrednost  $x$  nam je doduše nepoznata, ali ništa zato, dovoljno je da pogledamo koja je od sledećih vrednosti

$\text{sq}(r-1) + \text{sq}(y+1)$  i  $\text{sq}(r-x) + \text{sq}(y+1)$  „bliza“ vrednosti  $\text{sq}(r)$ . Zbog toga su i uvedene promenljive *left* i *right* koje figurušu u dvema sledećim naredbama. I nose u sebi odluku. Sada dolazi na red jedna pitalica i njena upotreba vrednost je sasvim jasna. Na kraju je potrebno još povećati promenljivu  $y$  i petlja je gotova.

I, eto, uspeši smo da se oslobodimo i kvadratnog korena, pa sve ostaje zatvoreno u polju celih brojeva. Ali, jedan pogled na tabelu vremena uveriče nas da u pogledu brzine još nije postignuto ništa značajno. Doduše, cilju smo se približili i ne treba biti nestrpljiv.

## Malo programiranja

Procedura *Circles6* vodi nas ka uspehu, bez obzira kakve kvalitete sama imala. Potrebne su joj izvesne obrade, ali suština će ostati ista.

U poslednjoj proceduri, uvedena je jedna promenljiva (*sq*) u kojoj se čuva kvadrat radijusa. Transformišimo izraze koji se dođeljuju promenljivim *left* i *right*:

$$\begin{aligned} \text{sq}(y+1) + \text{sq}(x-1) - \text{sq} &= \text{left} \\ \text{sq}(y+1) + \text{sq}(x) - \text{sq} &= \text{right, ili} \\ 2y+1-2x+1 + \text{sq}(y) + \text{sq}(x) - \text{sq} \\ 2y+1 + \text{sq}(y) + \text{sq}(x) - \text{sq} \end{aligned}$$

Umesto promenljive *sq* uvedemo promenljivu *correct*. Njena uloga će biti da čuva razliku  $\text{sq}(y) + \text{sq}(x) - \text{sq}(r)$ , a zato je i dobila ovo ime (*correct* = popravka). Ova razlika nastaje zbog toga što promenljive  $x$  i  $y$  predstavljaju samo približne vrednosti koordinata. Na ulasku u petlju promenljiva *correct* ima vrednost 0, jer je  $\text{sq}(r) + \text{sq}(0) - \text{sq}(r) = 0$ . Sada promenljive *left* i *right* dobijaju sledeće vrednosti:

$$\begin{aligned} \text{correct} + 2y+1-2x+1 &= \text{left} \\ \text{correct} + 2y+1 &= \text{right} \end{aligned}$$

Uloga promenljivih *left* i *right* je od ranije poznata. Promenljiva *left* predstavlja razliku

$$\text{sq}(\text{sq}(x-1), \text{sq}(y+1)) - (\text{sq}(x), \text{sq}(y+1)),$$

ili, što je isto,

$$\text{sq}(x-1) + \text{sq}(y+1) - \text{sq}(r).$$

Dakle, u pitanju je odstupanje koje se pravi kada se prave vrednosti koordinata  $(x, y+1)$  zamene parom  $(x-1, y+1)$ . Sa druge strane, promenljiva *right* odražava razliku koja nastaje prilikom aproksimacije parom  $(x, y+1)$ . Za to vreme promenljiva *correct* sadrži grešku, koja je učinjena prethodnom aproksimacijom parom  $(x, y)$ , i poslužila je da promenljive *left* i *right* dobiju prave vrednosti. Pogledajmo sada pitalicu koja sledi

```
IF abs (right) > abs(left) THEN BEGIN
  correct := left;
  x := x-1
END
ELSE correct := right;
```

Promenljiva *correct* sada preuzima jednu od vrednosti koje imaju *left* i *right*, u zavisnosti od toga za koju aproksimaciju smo se odlučili — za  $(x, y+1)$  ili za  $(x-1, y+1)$ . Sve tri promenljive: *correct*, *left* i *right*, sadrže iste „stvari“, tj. razlike nastale aproksimacijama. Jedino, neposredno pre pitalice, promenljiva *correct* sadrži razliku nastalu prilikom prethodne aproksimacije, a *left* i *right* sadrže razlike koje bi nastale primenom jedne od aproksimacija za sledeću tačku. Posle pitalice, promenljiva *correct* se pridružuje preostalim promenljivim, u „vremenskom“ smislu, tj. razlika koju sada sadrži odnosi se na istu tačku. Možda čitava stvar deluje malo mutno, ali da bi se dobro shvatilo potrebno je „prošetati“ algoritam i videti kako se menjaju pojedine varijable.

Procedura *Circles7*, koja je upravo opisana, delo je autora čiji identitet je zasada nepoznat piscu ovih redova. Očekujemo je da je u pitanju pravi majstor programiranja, jer je algoritam izuzetno maštovit, elegantan, dovoljno precizan i brz. Pri tom se svaki od ovih atributa može i dokazati.

S obzirom da imamo u rukama tako lep algoritam, bila bi velika šteta ne izvršiti određenu optimizaciju. I procedura *Circles7* predstavlja upravo zaokruženo celinu. Negde na početku ovog napisa rečeno je kako nikakve linije ne dolaze u obzir, ali evolutivno pomenuto izuzetka. Zbog nesavršenosti video čipova, naši krugovi neće biti baš okrugli. Praktično, sastojace se od vertikalnih i horizontalnih linija. Zato je u proceduri *Circles8* izvršeno razbijanje kruga na linije umesto na tačke. Osnova je pronađena u izvornoj prethodnoj proceduri. Ovo poboljšanje je omogućilo posmatranje promenljive  $x$ , a ona je velikim delom konstantna. Zato se ne crtaju tačke u svakom prolazu kroz petlju, već se, kada se varijabla  $x$  promeni, crtaju linije za prethodnu vrednost. Ostale promene su više tehničkog karaktera, i trebalo bi napomenuti još samo promenu uslova u pitalici. Sada umesto

$$\text{abs(right)} > \text{abs(left)}$$

stoji nešto jednostavnije

$$\text{left} > \text{right} > 0$$

Ova promena je pomalo sumnjiva, ali sasvim dobro funkcioniše, i, još više, može se strogo pokazati da vas nikada neće izneveriti.

Umesto zaključaka, pogledajte još jednom paskal program sa slike 1 i uporedite prvu i sedmu proceduru, pa će sve biti jasno.

## Za vlasnike „amstrada“

Procedura *Circles7* je prevedena na assembler i rezultat je bio zadivljujući brz mašinski. Priložen program predstavlja jedan RSX modul, a sadrži četiri naredbe. Pored naredbi za crtanje kruga, uključene su i rutine za elipse. Naredbe imaju sledeći oblik:

*I* CIRCLE, cent\_x, cent\_y, r, [pen]  
*I* ELLIPSE, cent\_x, cent\_y, r\_x, r\_y, [pen]

Argumenti *cent\_x* i *cent\_y* predstavljaju koordinate centra kruga. Argument *r* u naredbi *CIRCLE* predstavlja poluprečnik kruga, a *r\_x* i *r\_y* u naredbi *ELLIPSE* predstavljaju horizontalnu i vertikalnu poluos elipse. Parametar *pen* se koristi kao kod standardnih naredbi *DRAW* i *PLOT*, tj. za promenu pisaljke.

Pošto su kod „amstrada“ uvek zastupljene po dve naredbe za svaku grafičku rutinu, jedna sa apsolutnim parametrima (recimo *PLOT*) i jedna sa relativnim (*PLOR*), bilo bi ružno narušavati takvu harmoniju, pa su tu i sledeće dve naredbe:

*C*IRCLES, disp\_x, disp\_y, r, [pen]  
*E*LLIPSES, disp\_x, disp\_y, r\_x, r\_y, [pen]

Nije ih potrebno posebno objašnjavati, a razlika između ove i prethodne dve naredbe je ista kao i razlika između *PLOR* i *PLOT*.

Ovaj asemblerski program je vrlo strogo strukturiran i svaki red je brižljivo napisan. Iz ovih razloga ne bi trebalo da bude nikakvih zabuna u tumačenju značenja pojedinih delova programa. Ipak, tu i tamo naići ćete na poneki komentar (oni nikad nisu suvišni), koji će, nadajmo se, biti sasvim dovoljni.

Za one koji vole stvarno brze programe, tu je i jedan valjano napisan plot program. Može se povezati sa glavnim programom, a potrebno uputstvo je dato u komentaru na samom početku.

Dejan Muhamedagić



U prošlom broju započeli smo priču o matematičkim osnovama registriranja i obrade podataka temom brojni sistemi. Sada ćemo se pozabaviti načinima prikazivanja podataka u registrima računara. O ovom problemu već smo govorili u „Računarima 18“ [vidi tekst D. Ristanovića „Put u središte ROMA (10)“], ali pretpostavljamo da još jedan tekst na istu temu neće biti suvisan, jer ćemo je ovoga puta obraditi na način kako smo prikazali i brojne sisteme. Budući da je registriranje podataka posvećeno dosta pažnje u udžbenicima programiranja, daćemo samo kratak teorijski uvod i istorijski osvrt, a priloženi test, program generator zadataka i terminološki rečnik treba da doprinesu uspješnijem savladivanju gradiva. Prilog posvećen bibliografiji izostao je i ovog puta zbog nedostatka prostora. Knjigama i zbirkami zadataka koje obrađuju aritmetičke osnove posvećemo pažnju koju zaslužuju u narednom broju.

## Označeni brojevi

Postoji nekoliko metoda predstavljanja negativnih brojeva u računaru. Izbor konkretnog oblika registriranja značajno utiče na način realizacije aritmetičkih operacija. Zamislimo, za početak, da radimo sa dekadnim računarom. U uobičajenoj matematičkoj notaciji svakom broju prethodi znak + ili - koji odgovaraju pozitivnim i negativnim brojevima. U računaru bi praktičnije bilo da se za znak broja ne uvode posebni kodovi, već da se on reprezentuje nekom od cifara. Tako je uvedena konvencija da 0 na mestu za znak broja reprezentuje pozitivne brojeve, a najveća cifra brojnog sistema, u ovom slučaju 9, reprezentuje negativne brojeve. Tako bi, ako imamo na raspolaganju ukupno 3 dekadna razreda 019 bio zapis broja +19, a 919 zapis broja -19. Ako pak na raspolaganju imamo 4 dekadna razreda, onda bi isti brojevi bili registrirani kao 0019 i 9019. Princip se zadržava i u binarnom brojnog sistemu, s time što tamo negativne brojeve predstavlja i na vodećoj poziciji.

Tip podataka	Dužina	Opseg	
Integer	Označen	Neoznačen	
Byte	8 bitova	[-128, +127]	[0,255]
Word	16 bitova	[-32768, -32767]	[0,65535]
Longword	32 bita	[-2131, +2131-1]	[0,2732-1]
Quadword	64 bita	[-2763, +2763-1]	[0,2764-1]
Octaword	128 bitova	[-2727, +2727-1]	[0,2728-1]
<b>Floating point</b>			
F floating	32 bita	daje 7 značajnih	dekad. cifara
D floating	64 bita	daje 16 značajnih	dekad. cifara
G floating	64 bita	daje 15 značajnih	dekad. cifara
H floating	128 bitova	daje 33 značajne	dekad. cifre

Celobrojni i podaci u pokretnom zarezu VAX familije računara

Ovakav zapis označenih brojeva zove se **oblik znaka i apsolutne vrednosti**.

Međutim, zbog dva različita zapisa nule, zbog komplikovanog vršenja aritmetičkih operacija u ranim računarima, uveden je tzv. **oblik nepotpunog komplementa** u kome se kod negativnih brojeva svaka cifra broja zamenjuje svojom dopunom do najveće cifre brojnog sistema, a za zapis znaka broja važe ista pravila. Tako bi sa 4 dekadna razreda zapis brojeva +19 i -19 u nepotpunom

komplementu imali sledeći izgled: [+19]nk=0019, [-19]nk=9980. Zapis u nepotpunom komplementu omogućava jednostavnije vršenje aritmetičkih operacija, ali i dalje ostaju dva različita zapisa nule. (Ako radimo sa četiri dekadna razreda, imali bismo [+0]nk=0000 i [-0]nk=9999).

Zato se u savremenim računarima za zapis brojeva najčešće koristi **potpuni komplement** koji se za negativne brojeve dobije tako što se na mestu najmanje težine zapisa u ne-

## Terminološki rečnik

**BRJOG JE OZNAČEN** ako njegov zapis sadrži znak, u suprotnom je neoznačen.

Zapis broja u kome se na poziciju znaka nalazi 0 ako je pozitivan ili najveća cifra brojnog sistema u kome je broj zapisan, ako je negativan zove se **zapis u obliku znaka i apsolutne vrednosti**.

Zapis broja u kome se na poziciji znaka nalazi 0 ako je pozitivan, a ako je negativan umesto svake cifre stoji njena dopuna do najveće cifre brojnog sistema u kome je zapisan (komplement) i na mestu znaka najveća cifra, zove se **zapis u obliku nepotpunog komplementa**.

**Zapis broja u potpunom komplementu** isti je kao i zapis u obliku znaka i apsolutne vrednosti i obliku nepotpunog komplementa ako je broj pozitivan, a ako je negativan odbija se dodavanjem jedinice na mestu najmanje težine nepotpunog komplementu.

Broj je zapisan metodom **fiksno g zarezu** ako je pozicija decimalne tačke unapred određena, te se ne mora zapisati eksplicitno, već se podrazumeva.

**Značajne** cifre su sve cifre različite od nule i nula između značajnih cifara.

Zapis broja X u **pokretnom zarezu** podrazumeva da se broj predstavlja u obliku para (Xm, Xe), tj. X=Xm\*10<sup>Xe</sup>, gde je N osnova brojnog sistema u kome je zapisan X. Veličinu Xm zovemo MANTISA broja, a veličinu Xe **eksponent**.

**Normalizovana** mantisa je po apsolutnoj vrednosti uvek manja od 1 i ima prvu cifru iza pozicije tačke različitu od nule, a jednaka je 0 jedino za broj 0.

potpunom komplementu doda 1. Tako imamo: [+19]pk=0019, [-19]pk=9981.

Opseg celih brojeva koji se mogu zabeležiti u registrima računara primenom navedenih zapisa zavisi od broja bitova koji su na raspolaganju. Tako se kod mikroročunara za zapis celih brojeva obično koriste jedan ili dva bajta, ali se, recimo kod familije računara VAX, može koristiti 5 različitih dužina reči za zapis celobrojnih podataka (vidi tabelu u prilogu).

Primitimo da je interval brojeva koji se mogu predstaviti u potpunom komplementu nesimetričan u odnosu na 0, možemo zapisati jedan negativan broj više. Recimo, od 256 različitih zapisa koji se mogu naći u bajtu, 1 je rezervisan za zapis nule, 127 za pozitivne i 128 za negativne brojeve. U oblicima nepotpunog komplementa i znaka i apsolutne vrednosti ovi intervali su simetrični, ako koristimo jedan bajt imamo po 127 zapisa za pozitivne i negativne brojeve u bajtu, a onaj jedan zapis „viška“ utrošen je za zapis negativne nule.

Program koji vam ovog puta nudimo trebalo bi da pomogne u uvežbavanju prevodjenja celih brojeva u binarni brojni sistem i njihovog zapisivanja u registre računara. Izabrana je dužina od dva bajta koja se najčešće koristi u ove svrhe. Istina, onaj „pedagoški“ primer registriranja broja -0 nije specijalno predviđen pri generisanju podataka (generiše se samo+0), ali se može jednostavno umetnuti u program. Napominjemo da je pred vama samo deo programa koji ilustruje registriranje podataka. Registriranje u pokretnom zarezu i drugi načini zapisa numeričkih i nenumeričkih podataka biće posebno obrađeni. Uz potprogram za prevodjenje celih dekadnih brojeva u binarni oblik korišćen u prethodnom programu, tu su i potprogrami za formiranje nepotpunog i potpunog komplementa, oba rešena korišćenjem azbucnih funkcija (vidi bezik dodatak). Razume se, ovi problemi mogu se programirati i na drugačije načine. Na druga rešenja ćemo se osvrnuti na kraju priče o registriranju podataka.

Od prethodnog broja počeli smo seriju napisa Osnove programiranja namenjenih srednjoškolicima i njihovim profesorima. U njoj će biti obrađivane neke od nastavnih tema predmeta „Računari i programiranje“ s pristupom koji se razlikuje od onog uobičajenog u udžbenicima. Za svaku temu biće, uz kratak teoretski uvod i istorijski osvrt, priloženi GENERATOR ZADATAKA za uvežbavanje (algoritama i odgovarajućih bezik program). TEST koji treba da pokaže da li su usvojeni osnovni pojmovi nastavne teme. TERMINOLOŠKI REČNIK novouvedenih pojmova i BIBLIOGRAFIJA za sve one koji žele da prošire svoje školsko znanje. Nadamo se da će ovako koncipirani tekstovi moći da posluže učenicima kao dopuna i pomoć u savladavanju gradiva. Iako je strogo pedagoški usmerena, serija će nadamo se biti zanimljiva svakom mladom čitaocu koga interesuje programiranje.

## Vrlo veliki i vrlo mali brojevi

Do sada smo govorili isključivo o registovanju celobrojnih podataka. Međutim najviše se radi sa mešovitim brojevima, brojevima koji imaju i ceo i razlomljen deo. Oni se u registrima računara beže na dva načina: u FIKSNOM I POKRETNOM ZAREZU. Metoda fiksnog zarezu podrazumeva da je pozicija decimalne tačke unapred određena, te se ona ne mora zapisati eksplicitno, već se podrazumeva. U naučno tehničkim primenama uobičajeno je da se ovaj način koristi samo za registovanje celih brojeva, a metod pokretnog zarezu za sve



ostale. Međutim, u poslovnoj praksi nije tako. Računari predviđeni za komercijalnu obradu podataka rade najviše sa podacima zapisanim u fiksnom zarezu, bez obzira da li su oni celi ili ne, a pokretni zarez se koristi samo u izuzetnim slučajevima.

Zapis broja X u pokretnom zarezu podrazumeva da se broj predstavlja u obliku para  $(X_m, X_e)$ , tj.  $X = X_m \cdot 10^m X_e$ , gde je N osnova brojnog sistema u kome je zapisan X. Veličinu  $X_m$  zovemo MANTISA broja, a veličinu  $X_e$  ekspanent. Zahvaljujući ovakvom zapisu broja registujemo samo njegove značajne cifre, te ovakav metod služi za registovanje vrlo velikih i

## TEST 2. Registovanje podataka

1. U kom se obliku najčešće predstavljaju u računarima negativni binarni brojevi

- u nepotpunom komplementu
  - u potpunom komplementu
  - u obliku znaka i apsolutne vrednosti
2. Kojem od intervala pripadaju označeni celi brojevi koji se mogu zapisati u registru od 8 celija

- $[-127, +128]$
  - $[-128, +128]$
  - $[-128, +127]$
3. Koji je označen ceo broj upisan u registru  $R = [11111101]$
- +253
  - 3
  - 126

4. Kako glasi potpuni komplement broja  $b = -3910$

- 03910  $\rightarrow$  96089
- 03919  $\rightarrow$  96089 + 1 = 96090
- 93910  $\rightarrow$  06089

5. Kako glasi potpuni komplement binarnog broja -1011010

- 10100110
- 10100101
- 00100110

6. Šta je to predstavljanje brojeva sa fiksnim decimalnim zarezom

- predstavljanje brojeva u obliku reči sa fiksnim brojem mesta: faktor mere nije sadržan u reči nego samo mantisa
- predstavljanje brojeva u obliku reči promenljive dužine: reč sadrži podatak o faktoru mere i mantisi
- predstavljanje brojeva u obliku reči sa fiksnim brojem mesta: reč sadrži podatak o faktoru mere i mantisi

7. Koje se područje brojeva može predstaviti u brojnom sistemu sa osnovom 16 i brojem mesta  $s = 4$ , za slučaj predstavljanja sa fiksnim decimalnim zarezom

- $[0, B^s] = [0, 16^4] = [0, 65535]$
- $[0, B^s] = [0, 16^4] = [0, 65536]$
- $[0, B^s] = [0, 16^4] = [0, 64]$

8. Kojom se metodom predstavljanja brojeva najčešće radi kada su u pitanju uređaji za obradu podataka u komercijalne svrhe

- tehnikom predstavljanja brojeva sa fiksnim decimalnim zarezom
- tehnikom predstavljanja brojeva sa pokretnim decimalnim zarezom
- tehnikom sa fiksnim i pokretnim zarezom primenjuje se u

istoj meri za ovu svrhu

9. Zašto se ne radi isključivo tehnikom predstavljanja brojeva sa pokretnim decimalnim zarezom kada ovi obuhvataju mnogo veći opseg brojeva nego brojevi predstavljeni tehnikom pisanja sa fiksnim decimalnim zarezom

- nema uređaja za obradu podataka koji bi radili samo sa tehnikom pokretnog decimalnog zarezu
- tehnik predstavljanja brojeva sa pokretnim decimalnim zarezom mnogo je netačnija od metode predstavljanja sa fiksnim decimalnim zarezom
- Upravljanje računanjem mnogo je komplikovanije kad se radi sa pokretnim decimalnim zarezom, što ima za posledicu duže vreme računanja

10. Kako se zovu sastavni delovi broja zapisanog u obliku pokretnog zarezu

- mantisa i ekspanent
- mantisa i osnova
- osnova i ekspanent

11. Znak broja zapisanog u pokretnom zarezu određen je

- veličinom ekspanenta
- znakom mantise
- znakom ekspanenta

12. Od broja cifara mantise zavisi

- opseg brojeva koji se mogu predstaviti u pokretnom zarezu
- tačnost prikazivanja brojeva u pokretnom zarezu
- brzina izvršavanja aritmetičkih operacija u pokretnom zarezu

13. Mantisa broja u pokretnom zarezu zapisuje se

- u potpunom komplementu
- u obliku znaka i apsolutne vrednosti
- u nepotpunom komplementu

14. Zašto se za zapis ekspanenta koristi kod „višak 128“ ili „višak 64“ ili neki drugi kod „sa nivoom“

a) da bi najmanji ekspanent bio zapisan najmanjim binarnim brojem

- da bi se na mestu najveće težine našlo 1
- da bi se na mestu najveće težine našlo 0

15. Kojom vrednosti ima broj zapisan u pokretnom zarezu ako je mantisa normalizovana, ekspanent dat u kodu „višak 128“ i njihove vrednosti su  $E_x = 0000010$  i  $M_x = 000...00$

- 4
- 21 - 2
- 21 - 126

Odgovori: 1 - b, 2 - c, 3 - b, 4 - b, 5 - a, 6 - a, 7 - a, 8 - a, 9 - c, 10 - a, 11 - b, 12 - b, 13 - b, 14 - a, 15 - c.

## Azbučne funkcije

Pažljiviji čitaoci zapazili su da su u oba programa (za brojne sisteme iz prošlog i za registriranje celih brojeva iz ovog broja) korišćena rešenja pomoću azbučnih podataka (stringova). Kako upoznavanje sa azbučnim operacijama i funkcijama nije predviđeno nastavnim programima, a bejzici većine kućnih računara ih podražavaju pa bi bila šteta da se ne koriste, dajemo ovde njihov kratak pregled.

Dve osnovne operacije koje se vrše nad azbučnim podacima su spajanje i razdvajanje. Spajanje se u bejziku realizuje operatorom +, a za razdvajanje nije predviđen poseban operator već se realizuje pomoću funkcija LEFTS, MIDS i RIGHTS. Za rad sa azbučnim podacima koriste se i funkcije konverzije ASC i CHR\$, numeričke funkcije LEN i VAL i funkcija STR\$ koja daje odgovarajući „azbučni“ zapis brojeve promenljive.

**ASC. format:** ASC (<string>); **tip:** numerička funkcija  
**dejstvo:** proizvodi ASCII vrednost prvog karaktera iz niske <string>.

**primeri:** ASC(„A“) daje 65, ASC („BRAT“) daje 66, ASC ("" ) (prazne reči) trebalo bi da daje 0; (to nije slučaj kod komodorovih računara, ali ovaj bag se može prevazići upotrebom ASC (CHR\$(0)). Proverite na sledeći način PRINT CHR\$(0); „#“ i zatim PRINT ASC (CHR\$(0))

**CHR\$ format:** CHR\$ (<br. izraz>); **tip:** azbučna funkcija  
**dejstvo:** proizvodi jedan karakter čija je ASCII vrednost jednaka argumentu funkcije. Argument mora biti iz intervala [0,255]; ako je kao argument navedena vrednost u pokretnom zarezu, odbacuje se razlomljeni deo.

**primeri:** CHR\$ (65) proizvodi A, CHR\$(34) proizvodi navodnike, CHR\$(13) proizvodi prelazak u sledeći red, a boja i inverzno prikazivanje takođe pomoću ove funkcije mogu da se kontrolišu na ekranu.

**LEFTS. format:** LEFTS (<string>,<br>); **tip:** azbučna funkcija

**dejstvo:** izdvaja prvih <br> levih karaktera iz azbučnog podatka <string>; argument <br> mora biti ceo broj iz intervala [0,255]; ako je taj broj veći od dužine reči, uzima se ceo podatak, a ako je 0 uzima se prazna reč.

**primer:** LEFTS („ANANAS“), 3) daje „ANA“. LEFTS se često koristi za pozicioniranje kursora: reč sa kursor kontrol karaktera na koju se primeni ova funkcija u PRINT naredbi omogućava da se kontrolišu pozicija kursora na ekranu; na primer:

10 AS="CRSR=>CRSR=>... CRSR=>  
20 PRINT LEFT\$(AS, 10)

pomera kursor na ekranu udesno za deset mesta.

**LEN. format:** LEN (<string>); **tip:** celobrojna funkcija

**dejstvo:** izračunava broj znakova u podatku. Blanko pozicije i negrafički simboli se takođe uračunavaju.  
**primer:** LEN („ANANAS“) daje 6.

**MIDS. format:** MIDS (<string>), <br>1>, <br>2>; **tip:** azbučna funkcija

**dejstvo:** izdvaja podniku iz reči <string> čiji je početak definisan argumentom <br>1>, a dužina argumentom <br>2>. Dozvoljene vrednosti za <br>1> su između 1 i 255, a ako se navede broj van ovog intervala dobija se poruka ? ILLEGAL QUANTITY ERROR. <br>2> može biti iz intervala [0,255], a ako je van njega dobija se ista poruka greške, ako je prvi brojni argument veći od dužine stringa rezultat je prazna reč, a ako se izostavi drugi argument uzima se čitav preostatak. MIDS(SS, 1, 1)<=>LEFTS(SS,1)

**primer:** MIDS („BANANA“, 2, 3) DACE „ANA“.

**RIGHTS. format:** RIGHTS (<string>), <br>); **tip:** azbučna funkcija

**dejstvo:** izdvaja podreč koja se sastoji od desnih <br> znakova azbučnog podatka <string>. <br> mora biti iz intervala [0,255]; ako je 0 dobija se prazna reč.

**primeri:** programski red

AS="1234567890";L=LEN(AS);FORI=1TOL:PRINT  
LEFT\$(AS,I)+RIGHTS (AS,L-I);NEXT  
daće niz kopija reči AS. Kako tako rade funkcije LEFTS i RIGHTS možete videti ako u prethodnoj programskoj liniji modifikujete naredbu PRINT na sledeći način: PRINT LEFT-\$(AS, I), RIGHTS (AS,L-I).

**STR\$ format:** STR\$(<br>); STRz(<br>); **tip:** azbučna funkcija

**dejstvo:** daje azbučnu reprezentaciju argumenata te se koristi prilikom formatizacije štampe.

**primer:** PRINT STR\$(1234500000000) i PRINT STR\$(.0000000000000000) proizveće 1.2345E+12 i 1E-15

**napomena:** ova rutina ubacuje vodeće blanko praznine pa bi LEN (STR\$(1)) dalo vrednost 2.

**VAL. format:** VAL (<string>); **tip:** numerička funkcija  
**dejstvo:** izračunava numeričku vrednost koja je data <string> argumentom; ako prvi karakter nije +, -, blanko ili cifra odgovarajuća vrednost je 0. Konverzija stringa se završava kada se nađe na znak koji nije cifra (ili decimalna tačka ili oznake E za exp.)

**napomena:** ove funkcije mogu se realizovati i na računaru „spektrum“, ali imaju drugačiji sintaksu.

vrio malih veličina koje izlaze iz opsega brojeva koji se mogu predstaviti u obliku fiksnog zarez. (Značajne cifre su sve cifre različite od nule i nula između značajnih cifara. Na primer, u brojevima 12000 i 0.00012 značajne su samo cifre 1 i 2, a u broju 102000 značajna je i nula između cifara 1 i 2.) Da bi zapis brojeva u računaru bio jednoznačan, koristi se *normalizovana* mantisa (ona koja je prvi razlomak čija je prva cifra iza pozicione tačke različita od nule — u binarnom sistemu to mora biti 1, ali u dekadnom 9 različitih cifara mogu da se nađu na poziciji 10<sup>-1</sup>—1)

Tragovi aritmetike pokretnog zarezu vode nas do starog Vavilona, gde je oko 1800 godina p.n.e. bio korišćen brojni sistem osnovne 60 u kome su operacije vršene u pokretnom zarezu. Pri tome se odgovarajući eksponent nije pisao, već

ga je „podrazumevao“ onaj koji je računao.

Drugi primer ranog korišćenja veličina u pokretnom zarezu povezan je sa imenom grčkog matematičara Apolonija (III vek p.n.e.), koji je po svemu sudeći bio prvi koji je objasnio kako se može uprostiti množenje, sabirajući stepene 10 izdvojeno od njihovih koeficijenta.

Otkriće logaritama 1600. i ubrzo zatim logaritama (1630.) pomogli su da o ovaj način zapisa brojeva bude široko prihvaćen.

Jedna od prvih sugestija da se pokretni zarez koristi u budućim računarima može se naći u radu Leonarda Toresa (L. Torres, (1852.—1936)), španskog naučnika i pronalazača, objavljenog 1914. Tricest godina kasnije ovo je i realizovano na relejnim mašinama Model V (Bell Laboratories) i hardverskom Marku II. Štbić je još 1939. predložio hardversku rea-

lizaciju pokretnog zarez za računsku mašine firme Bell Laboratories. Otrpille u isto vreme korišćen je pokretni zarez i u mašinama koje je projektovao K. Cuze u Nemačkoj.

Danas je po mnoštvu „manuela“ moguće pronaći detalje o programima za rad u sistemu pokretnog zarez, a mi ćemo se zadržati samo na najčešće korišćenim rešenjima.

Za zapis mantise u pokretnom zarezu obično se izdvaja tri do četiri bajta u kojima se ona piše u obliku znaka i apsolutne vrednosti. Broj cifara mantise utiče na tačnost registrovanja brojeva. Tako, ako za nju izdvojimo tri bajta možemo računati na 6 značajnih dekad-

nih cifara, a ako izdvojimo četiri dobijamo 9 značajnih dekadnih cifara rezultata. U prilugu možete videti kako je rešeno registrovanje brojeva u pokretnom zarezu kod familije računara VAX, a kod većine mikro-računara brojevi se zapisuju u tzv. petobajtnoj formi, jedan bajt je predviđen za eksponent i četiri za mantisu.

Prilikom određivanja veličine eksponenta treba imati u vidu u kom se brojnoin sistemu zapisuje broj. Sam eksponent se, doduše, zapisuje u binarnom obliku, ali se u nekim rešenjima podrazumeva da se broj i mantisa beleže u heksadekadnom sistemu. Po određivanju eksponenta on se zapisuje u obliku potpunog komplementa uvećanog za 1 na poziciju znaka eksponenta (kod „višak 128“ kod mikroručunara).

Podacima u pokretnom zarezu i vršenjem aritmetičkih operacija nad njima bavićemo se više u sledećim brojevima.

## Nagradni zadatak

Odredite prvi nekoliko cifara (više od 10) broja 1986. 1987.

Zadatak predložio Dragan Kulezić





Šta ima novo

**Stabilni izvor referentnog napona**

Većina metoda analognog/digitalne konverzije, kakve se primenjuju i u personalnim računarima, zahteva stabilan i precizan izvor referentnog napona. Zato firma Analog Devices nudi AD588 monoilitni izvor referencije čija inicijalna greška iznosi 1 mV, a stabilnost je bolja od 1,5 ppm/°C, (1 ppm = 1 milioniti deo). Ovo integrirano kolo ima programabilne izlaze od +10, -10, +5, -5, ili ±5 V. Navedena stabilnost i 12-bitna tačnost je ostvarena primenom tzv. Kelvinkontakata.

**Svestrani Modem**

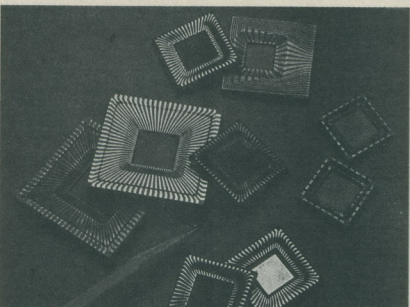
Američka firma Rockwell International, odnosno njen deo zadužen za elektronske komponente, proizvodi interesantan modemski modul, na Evropa kartici, koji uključuje i brzinu rada od 9600 bps (bitova po sekundi), pored već standardnih 7200 i 4800 bps. Osim što se slože sa CCITT V.24 standardom (evropski standard kao RS232), modem pod oznakom R96FT/SC takođe zadovoljava i standarde CCITT V.29 i V.27, a ima i tzv. sekundarni kanal od 75 bauda.

**Siemens-ova grafika**

Renomirana nemačka firma Siemens mnogo očekuje od svojih, novog grafičkog kontrolora pod oznakom 95C60 opisno nazvanom QPDM. Ovaj LSI čip omogućuje integraciju grafike i teksta na bit-mapiranom grafičkom sistemu, crta vektore sa brzinom od 120 000 vektora/s, pokazuje tekst sa brzinom od 50 000 znakova/s, podržava ekrane sa rezolucijom od 2000 x 2000 tačaka, a popunjava poligone sa vremenom od samo 50 ns po pikselu (pixel = element slike kod grafičkih terminala). Uz radnu frekvenciju od 20 MHz, 95C60 QPDM može da radi sa 256 memorijskih ravni i može se integrirati lako u postojeće 8/16/32 bitne sisteme, pa će verovatno naći primenu u CAD/CAM radnim stanicama.

**Mudri Thomson**

Investicija koju je poznati francuski koncern Thomson uložio u visokotehnološku američku firmu Mostek počinje da se ispla-



ćuje. Naime, Thomson Semiconducteurs, deo koncerna zadužen za poluprovodničke komponente, već proizvodi veoma brze (tačnije, koliko je autoru poznato najbrže) MOS statičke RAM memorije (CMOS tehnologija). RAM-ovi sa oznakom MK41H48 imaju 4K x 4 organizaciju, DIL 20 kućice, mirnu struju od samo 50 mikroA, a kompletno vreme za jedan ciklus iznosi samo 20 ns. Model pod oznakom MK41H69 ima vreme pristupa od samo 10 ns, dok kolo MK41H78 radi sa istom brzinom i ima zaseban pin za kontrolu izlaza (output enable). Svi memorijski modeli rade pri napajanju od samo 5V, a nudi se i JEDEC verzija za niskonaponski rad (tzv. LV-TTL 3.3 V verzija).

**Jedinstveni čip za grafiku**

Američka firma Paradise Systems preporučuje svoje rešenje za grafičku podršku IBM i kompatibilnih računara: ono se sastoji u upotrebi jednog kompleksnog čipa pod oznakom PEGA2, koji podržava IBM EGA format (Enhanced Graphics Adaptor) sa rezolucijom od 640 x 480 linija (horizontalna frekvencija od 31 kHz), odnosno rezolucijom reda 640 x 400 linija (hor. frekvencija od 25 kHz). Čip je kompatibilan i sa drugim formatima kao što su CGA (Colour Graphics Adaptor), MDA (Monochrome Display Adaptor) i Hercules monohromatskom grafikom. Ovaj čip se lako povezuje sa računarima koji su bazirani na Intelovim procesorima tipa 8086, 8088, 80286 i 80386.

**Ekonomični Fujitsu**

Obzirom da proizvodi Intelove komponente po tzv. „second source“ sporazumu, japanska firma Fujitsu se pobrinula da olakša rad konstruktorima sistema sa 8086 mikroprocesorom. Stoga se nudi integrirano kolo pod oznakom MB8939 koje u sebi integriše generator takt frekvencije, BUS kontroler, kontroler prekida/interapta, kontroler za direktan pristup memoriji (DMA), a i vremensko kolo/tajmer, sve na jedinstvenom čipu. DMA kontroler ima četiri nezavisna kanala. Očevidno je da ovo kom-

pletano CMOS LSI kolo zamenjuje bar 5-6 sličnih integriranih kola.

**Proboj kod DRAM-ova**

Dok poznate firme kao Toshiba, Fujitsu, Siemens, Intel i dr. nude dinamičke memorije sa kapacitetom od 1 Mbit (organizacija je 1 Mbit x 1), američki Texas Instruments proizvodi već uzorke dinamičke memorije sa kapacitetom od 4 Mbita! Ovo integrirano kolo je izrađeno CMOS tehnikom, sa linijama preciznosti 1 mikrometar, i sadrži preko 8 miliona aktivnih komponenti na jednom čipu površine oko 1 cm<sup>2</sup>. Konfiguracija je nešto praktičnija i iznosi 1 M x 4 (nibble output), a vreme pristupa je 170 ns od RAS (Row Address Strobe) signala, dok je vreme dekodiranja ispod 30 ns za SCD (Static-Column-Decode). Proizvodnja u većim količinama (tzv. „volume production“) se očekuje tokom 1987. godine.

**Još jednom Thomson**

Ne samo što proizvodi kompletnu 68xxx familiju komponenti, ne samo što nudi najbrže statičke memorije u ovom trenutku, ova čuvena francuska firma proizvodi celu familiju programabilnih kompleksnih signalnih procesora (čipovi koji su namenjeni za obradu analognih signala na digitalan način, jednostavno rečeno). Tako npr. integrirano kolo sa oznakom TS68930/31 predstavlja 32-bitni mikrokomputer za obradu signala koji ima ekvivalentni broj od 6,25 MIPS (miliona instrukcija u sekundi).

**ZYMOS na sceni**

Komplet od dva čipsa, sa oznakama 60044/60045, može da zameni praktično sve kola sa IBM PC kompatibilne matične ploče! Potrebno je dodati, naravno, CPU tipa 80286 za AT model, i 8042 kontroler za tastaturu, kao i određen kapacitet memorije.  
Priprema: **Blažimir P. Miše dipl. ing.**

# Razbarušeni sprajtovi

Uređuje: Vlada Stojiljković



HERCULES-POKE2740, 75; POKE2741, 210; POKE2742, 10  
MR. DO'B CASTLE-POKE2371, 159; POKE2372, 255  
TOM AND TERRA-POKE2695, 234  
GORF-POKE7034, 234; POKE7035, 234  
SPACE ACTION-POKE34911, 0  
GATEWAY TO AFBGAI-POKE331, 234  
MAGIC CARPET-POKE33759, 234  
MR. ROBOT-POKE11555, 274  
SUPER PIPELINE-POKE3392, 234  
ACTION BIKE-POKE13119, 234 (ZA BEZBROJ ŽIVOTA)  
ACTION BIKE-POKE13822, 234; POKE13825, 234; POKE13835, 234 (ZA GORIVO)  
UGH!-POKE30319, 240  
DUTSBERT IN THE JUNGLE-POKE19243, 234; POKE19244, 234  
BMX RACERS-POKE2607, 274  
NEW YORK CITY-POKE19362, 162 (POSTAJETE NEUNISTIVI)  
E.C. BILL-POKE42934, 162  
ALIEN ATTACK-POKE20908, 162  
SAUCER ATTACK-POKE22044, 234 (ZA VEĆE METKE)  
LORD-POKE26901, 6  
JAMMIN-POKE20746, 255  
ASTROBLITZ-POKE36932, 162  
MEPHISTO-POKE25922, 173; POKE26470, 173

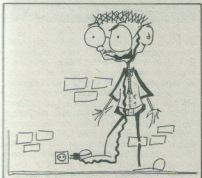
## LOAD „ja“

Svi mi — ili gotovo svi — nosimo negde u dnu glave bar jednu igru koja ne postoji. Neko je ima čitavu, razradenu do detalja, a neko samo skiciranu, ali svi bismo sutra seli i počeli da je pišemo.

Kad bismo znali kako.  
U tome i jeste čvor: znalci su retki. Naš san nema izgleda da postane java. Krivo nam je zbog toga.

A možda... kad bi postojao način... kad bismo mogli nekako svoju ideju da saopštimo znanцу... možda bi on...

Ova rubrika je upravo zato. U njoj će vaše ideje o igrama izlaziti na svetlost dana. Stvorice se šansa da ih vide sposobni programeri. „Bonzin vaš — ideje naše“, što rekao Ostop Bender; može se lako desiti da u taj način zamišljene igre postanu stvarne.



Onda bi nam svima bilo milo.

I evo, počinjemo. Sinopsis Stjepana Zlodija iz Kravarskog (dopunjen ilustracijom), podrugljiv komentar na mikro- i makro-svet u kome živimo, mogao bi biti podloga za efektanu, pamtljivu i, nadasve, originalnu igru. Hoće li je neko i napisati?

## Samubojica

Osnovni cilj ove igre nije ubiti nego — biti ubijen. Koliko mi je poznato, ovo je jedinstven slučaj u povijesti kompjuterskih igara i zbog toga očekujem da ova igra izazove revoluciju u proizvodnji softverskih limunada. (Dakle, kod nas je revolucija već počela, a zapadne zemlje mogu tek kaskati za nama. To je, također, bez presedana u povijesti dječvika otkad je izumio konja. Prije, naime, nismo mogli ni kaskati.)

Igra se sastoji od sedam nivoa. Skrin se sastoji od dva dijela. Gornji je rezerviran za

vašu oštrumu akciju. Donji dio služi da prikazuje vašu životnu energiju. Ona ili pada ili raste zavisno od situacije. Pada kad dodirnete predmet s mrtvačkom glavom. Na nulu pada tek kad izvršite cijeli projekt za samoubojstvo na jednom od nivoa na kojem se nalazite. Postoje i određene stvarčice koje vam povećavaju energiju. (To su sve one stvarčice koje ne spadaju u grupu spomenutu koji red prije. Predmeti za izravnu upotrebu ne spadaju ni u jednu grupu, a djeluju samo kad su svi na okupu.) Kad energija dosegne 100%, vraćate se na prethodni nivo.

Na početku se na ekranu pojavljuje tekst:

„Vi ste vrlo human čovjek i zbog toga se želite ubiti. O čemu se radi?  
Ostavili ste svoje nasljedstvo grupi štakora stanovnika zagrebačkog metra. Kako nema metra, njima prijete izumiranje i vi želite da oni što prije mogu dobiti nasljedstvo i izgraditi metro, pa se želite ubiti. U tome vas želi spriječiti dobra vila koja vas voli. Dakle, sve što vas sprečava to je njezino maslo...“

### Prvi nivo

Trebate pjesmočnu lupati po tastaturi sve dok se vaš lik (mali, crni s velikim naočalama) ne umori trčeci po beskonačnu trim-stazi i ne ispusti dušu. Osim ostalih, već najavijenih smetala, pojavljuje se jedna velika opasnost — djevojke s trenirkama „apple“. Kad vas jedna takva uhvati, energija (i ne samo energija) vam skače na 101%, a onda zna se što.

### Drugi nivo

Pokupite televizor, naslonjač, video-rekorder i 115 kasetna sa natpisom „Dynasty“. One bez natpisa ne dirajte, jer sadrže izvjesne filmove koji bi djelovali vrlo okrepjujuće na vaš moral, a samim tim i na energiju. Dakle, kad pokupite potrebne predmete, zavalite se u naslonjač, upalite televizor stavite kasetu i umрите od dosade!

## ČIP U RAČUNARIMA

u saradnji sa radio-emisijom „Čip i sedam jarica“ koja se emituje svake subote na talasnoj Radio-Beograda u 14 časova

### Treći nivo

Sastavite „Čelendžer“. Za ostalo će se pobrinuti NASA.

### Četvrti nivo

Najvažniji dio na ovom nivou je front gdje se bore dvije vojske. Vaš zadatak je sljedeći: nadite i uzmete bijelu zastavu, crnu boju, kist... Na zastavi napišite MIR i zaletite se između dvije vatre. Sekundu kasnije postali ste švicarski sir. Ako se napravite pametni i na bojno polje dođete sa čistom bijelom zastavom, ili je noć, vatra će prestati, a vila će vas prebaciti na prvi nivo.

### Peti nivo

Nadite C64 i disketnu jedinicu 1571 te ploču novokomponovane narodne muzike. Sve propisno priključite i stavite ploču u drav... Na ovom nivou postoji jedna velika prepreka. Okolo šeće Amiga i ukoliko vas uhvati postajete besmrtni... Vila vas vodi na vjenčanje i to je kraj... Ali, ako vas ne uhvati...

### Šesti nivo

Morate uhvatiti kinka koji nosi „spektrum“ i oduzeti mu ga... Priključite ga na televizor („spektrum“, a ne kinka) i malo pročitajte. Kad se vratite, primijetite čete da se „spektrum“ zagrijao od crvenog usijanja. Sjednite na njega...

### Sedmi nivo

Trebate sagraditi nuklearnu elektranu. Kad napokon ubacite uran u reaktor, prikačite dok zasvijetli žarujka koju ste priključili, uništite vodovod (koji ste opet vi sagradili) i prikačite trenutak... Preko cijelog ekrana će se prikazati mrtvačka glava koja će se modificirati u karakterističnu gljivu, a ova će se uz zaglujujući BIP (u ovoj pogodnosti će uživati samo vlasnici „spektruma“) pretvoriti u slova:

mrtav si  
MRTAV SI  
mrtav si

Dobra vila te nije spasila!!!  
I ona je mrtva!!!  
Ona se naime hranila povrćem  
koje si ti tako bezdušno ozračilo!!!  
VIDIMO SE U PAKLU!



## RANDOMIZE GAME

Jedan Dorčolac, sinkierista, navirao je 556.000 poena igrajući **Bomb Jack**. Možemo li to, do daljeg, smatrati rekordom? Ili je neko odavno nadmašio Dorčolca?

Ako jeste, da li je stigao do miliona? Pitanje se postavlja ne zato što je milion velik i okrugao broj, nego zato što **Bomb Jack** — igra koja se proizvodi od beskraja — ima samo šest cifarskih mesta. Da li se posle 999.999 broja vraća na nulu? Može li se u tekstualnom izveštaju napisati sedmoćifren broj? Ili se kod šest devetki igra ipak završava?

Grupica usmerenjaka pisala je avanturu; pisala pa zabetalila. Koliko znam, neće je dovršiti.

Šteta, jer ta je igra u našem blizu korak napred. Znajući da frustrirani i srditi igrači hoće da ukucaju i pokuju skarednu komandu, autori su za skoro svaku pripremlili odgovor („...Šar to bilo“, „...Kakav je to rečnik?“, „...Zar te tako majka vaspitala?“ itd.). To rešenje, po meni, sadrži lepu meru ljudske topline u poređenju s onim uobličajenim, nemaštovitim, hladnim i uvek istim „Ovu reč razumem“ ili već tako nekak. Primećujemo, uz to, i naglašen smisao za čujnost.

Ko kaže da je usmereno obrazovanje udaljeno od života?

Opaske jednog igrača su podložim stažom (iznesene ovde u skraćenoj obliku):

„Pogledajte životinje u igrama. Isprva su se pojavljivali insekti (pauci, stonoge), a kako je vreme odmicalo, tako su nailazili vodozemci (žabe), pa gmizavci (zmije), pa ptice (kondori), pa sisari (konji, psi, čovekoliki majmuni). Lepo se vidi da njihova evolucija sledi prirodnu. Redosled nastajanja vrsta i rodova isti je u igrama kao i u stvarnom životu.“

Jedan od njegovih zaključaka: „Da su kompjuterske igre postojale u doba inkvizicije, programere bi spaljivali žive.“

Nemam komentara.

Pirate neko hvali, neko kudi, a neko trpi. Ja im zameram nedostatak mašte.

Kad već razbijaju igre, što ih ne bi i prilagođavali domaćim uslovima? Sve-kako bi nam **Skool Daze** bio bliži i draži ako bi se, umesto one kazne „PREPIŠI 100 REDOVA“, na ekranu ispisivalo: „KEC IZ VLADANJA“ ili već tako nešto. Profesori bi mogli govoriti: „SEDI DRVO NA DRVO!“, a bonus ili uz „DANAS GUBITE MATEMATIKU“.

Zamišljam, dalje, **Stop the Train** sa dopunskom priručnjakom (za zakašnjenje), i komadom LET Z = 30 + INT (RND \* 300), s tim što bi se na ekranu povremeno pojavljivalo obaveštenje da voz kasni s minuta.

Bio bi dobar i **Impossible Mission** gde bi lift (opet na RND) bio u kvaru. Ili **Dynomite Dan II** sa novokomponovanim narodnjacima. Ili **Match Day** sa opcijom PODEMITTE SUDJU.

Mogućnosti su neograničene.



## Gotovo je, gotovo!



## ACE OF ACES

U ulozu engleskog pilota iz II svetskog rata, vi upravljate znamenitim dvomotornim Mosquito i uništavate neprijateljske projektile i ratnu opremu. Kad startujete igru, na ekranu će se pokazati meni koji vam daje da birate između vezbanja (PRACTICE) i obavljanja misije (MISSION). Ako izaberete ovu drugu opciju, možete sami sebi odrediti misiju kombinujući 4 zadatka. To su: V-1 ROCKET (sprečavate zloglasne teledirigovane rakete „fau-1“ da unište London), BOMBER (napadate grupu bombardera), TRAIN (uništavate voz) i U-BOAT (uništavate neprijateljsku podmornicu). Za početak, dok se ne uigrate, najbolje je odabrati prvu i treću misiju.

Kad izaberete misije, kompjuter će vam dati podatke o metama, vremenu, potrebnoj naoružanju i visinama na kojima lete neprijateljski avioni. Dobićete i mapu oblasti nad kojom ste se našli. Tada se naoružajte maksimalnim tovarom bombi i mitraljezjske municije, pa na posao.

Na raspolaganju su vam 4 ekrana i mapa koju menjate na tastaturi brojevima 1, 2, 3, 4 i 5. Prvi i glavni ekran predstavlja pogled iz pilotske kabine unapred. Ovde imate kompas, brzinomer, visinomer, radar i instrument koji pokazuje odakle neprijatelj

BRZAK	TRAJ.	BRZOST	TRAJ.	BRZAK	TRAJ.	BRZOST	TRAJ.	
1	W	240	200	170	A	140	130	70
2	W	250	190	150	B	180	120	60
3	W	260	170	100	B	160	110	60
4	W	80	40	19	7	140	60	50
1	I	220	150	120	6	130	70	40
2	I	210	190	90	9	120	60	30
3	I	200	140	60				

TRAJANJE JAKOŠNE UGRADBE SE IZRAČUNAVA GOLF I LI I IZRAČUNAVA U JEDNIMA I

TRAJANJE IZRAČUNAVA

nalazi. Drugi i treći ekran, daju vam pogled na levu i desnu stranu. Na njima možete povećavati brzinu, uključivati zakrlica i, ako treba, isključivati motore. Poslednji ekran prikazuje prostoriju u kojoj su bombe. Odavde, naravno, bombardujete.

Najpre na prvom ekranu navedete avion na sredni kurs (crta na kompasu mora doći na praviku); malo kasnije, na ekranu će se pojaviti rakete. Pridite im polako, uništavajte ih jednu po jednu, mitraljezjskim rafalima. Za svaku novu raketu morate odrediti nov kurs, zato se oslonite na radar u donjem desnom uglu: na njemu zelena tačkica označava neprijatelja. Kad uništite sve rakete, uzmite nov kurs i spustite se na ispod 1000 fita (stopa). Tada predite na drugi ekran i otvorite kapak na ležištu bombi. Ubrzo ćete kroz otvor ugledati voz — i nišan, koji možete pomerati. Cilj vam je da pogodite bar jedan vagon. Potom uzmite nov kurs (ponovo dovedite iglu kompasu u sredinu) i sačekajte na mapi dok ne stignete do Londona: tada će vam kompjuter čestitati i odrediti vam čin.

Tokom cele igre napadaju vas neprijateljski lovci: oni su veoma brzi i mogu vam uništiti neki od instrumenata. Pazite se, jer ako vam pogode kompas, sve je propalo.

U preostalim dvema misijama imate praktično iste ciljeve — samo se scenografija menja.

Ivan Albreht



Miloša Gašić

Evergreen

## IMPOSSIBLE MISSION

Ne znam zašto. Možda zato što je teška. Teška ne u smislu da su komande preosetljive ili da ima hiljadu ekrana, nego u tome što se od mene traži da koristim i ruke i glavu. Treba misliti, ali misliti brzo.

Postoji i element sreće, kao neka mala lutrija. Neki put se sav polinom da stignem do nekog komada nameštaja, a kad ga ispitam, kompjuter javi: „Nothing here“. Drugi put, opet, u prvom ili drugom pokušaju, bez muke, nađem šta mi treba.

Grafika takođe. Vidim da programer nije zalio trudu. Figura je animirana kao u crtanom filmu, milina je gledati. U drugim igrama figure su obično uprošćene preko mere, sve neki pravougaonici na dve noge.

Efeti su dobri, ali glas je najbolji od svega. To sam prvi put čuo sintetizovan glas. Odlično urađeno!

## evergreen/nevergreen



Nevergreen

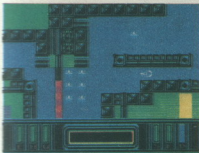
## COMMANDO I sl.

Mogu da pućam u životinje, u čudovišta, u brodove sa drugih planeta, u asteroide, ali u ljude? Ne mogu, pa ni u igri. Još je gore kad je to ubijanje najglavniji cilj, kad treba ljude tamariti kao garnad. Ne razumem kako neko može da nalazi zadovoljstvo u tome. Ko zna kakvi se tu kompleksni uživavaju. Kakvi to ljudi uživaju u nasilju?



## XARQ

ZARK  
Electric Dreams



Ploveće ostrvo? Ništa novo, ima toga još kod Žila Verna, da i ne pominjemo njegove SF-naslednike.

Ali ovo je veštačko, i toliko je pametno da same sebe da nadograđuje — a to mu se dopušta! ne sme, jer ako naraste i prekrije celu površinu planete Zargon, onda ode mast u propast.

Intervenirajte. Imate brz motorni čamac, lasere, dubinske bombe, dirigovane projekte i minobacače; to će vam valjda biti dovoljno da razvitate odbrambeni sistem Zarka, prodrete do centra i uništite glavni reaktor. Laseri pucaju pravo, a dubinske bombe puštate čim čujete signal na sonaru (signal znači „podmornica u blizini“). Projektlima i minobacačima morate odrediti nišanu ku daljinu pre palje; razlika među njima u tome je što su projektili dirigovani (tj. možete ih navoditi), a mine lete strogo po zakonima balistike.

Da biste napredovali, uništavajte brane: tako će voda preplivati izolacione robove, pa će po njoj vaš čamac moći da pliva do sledećeg nivoa. Budite spremni na sve, jer ste izloženi napadima sa tla, iz vode i iz vazduha. Nezgodna je veštačka inteligencija: kad nanjuši ljudsku.

## DARK SCEPTRE

Tamni skiptar  
Beyond



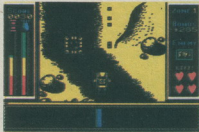
Sprotnosti se privlače i verovatno je zato toliko mnogo kompjuterskih igara sa vilama, vešticama, zlim volšebnicima, činjama, čudovištima i, uopšte, sa mitološkim i legendarnim inven-

tarom. U takve igre ulazite — primetili ste — ne dok su tekli med i mleko, nego kad se dogodila neka krupna nevolja, i stalno moramo nekoga da vadimo iz bule.

Ovog puta tragamo za tamnim skiptrom, jer je u njemu sva moć zlih osvajača. Srećom, nismo sami: biramo ekipu. Dato nam je nekoliko likova na raspolaganju (uglavnom su jedni vičniji borbi, a drugi mozganiji), a voda im je, iliti starešina klana; bez njega ne možemo. Na neredu, naše borbe može zavrbovati i neprijatelj, što obično otkrivamo kad je već kasno.

Sve je u strategiji; tačnije: u komandama koje izdajemo birajući ih u meniju. Moramo analizirati situaciju, predviđati neprijateljevu reakciju, uskladičiti dejstvo članova ekipe i paziti kome šta naređujemo (besmisleno je, na primer, tražiti od lovca da baca činj). Moramo, uz to, imati i malo sreće i mnogo, mnogo, mnogo vremena.

## STAINLESS STEEL



Nerdajući čelik  
Mikrogen

Neki put je to bio brod, neki put avion, neki put običan automobil (već zavisi od toga kako se igra zvala), ali priča je, u suštini, bila ista: pucali ste i vrdeli, vrdeli ste i pucali. Igra je od vas tražila brze reflekse i nešto malo taktičkog umeća; ništa više.

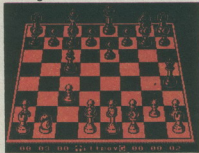
Istorija se ponavlja: opet ste u toj situaciji. Neprijatelj su ovog puta androidi, a njihov zli vođa ovog puta se zove doktor Varos. Do njega stižete kad (i ako) prodete kroz četiri zone; u svakoj, naravno, morate počešiti sve što je neprijateljsko, pa tek onda razmišljati o prelašku u nov ekran. Pri tom nema mnogo milosti: vreme vam je ograničeno, a androidi pucaju pa ne znaju šta je doista.

Jasno, imate i treću brigu: gorivo. Nailazite mestimicno na kanistere sa visokootkanskim benzinom ili već tako nečim; pripazite kako ga trošite, energetska kriza zahvatila je i kompjuterske igre.

Čuvajte džojstik: on u ovakvim igrama najgora prolazi.

## PSI CHESS

Pal-šah  
The Edge



Nagledali smo se šahova u stotinu verzija, i prva nam reakcija može biti: „Jo, zar opet?“

Ali ne bi trebalo da bude. Svaka nova verzija, da bi preživela, mora doneti neku novinu, neko poboljšanje. I ovde je takav slučaj.

Recimo, možemo birati između dvodimenzionalne i trodimenzionalne slike. U prvoj su uobičajeni simboli za figure, a u drugoj opet izbor: figure klasično dizajnirane i figure razrađene sa više stvaralačke ambicije. I poseti se mogu učiti dvojak: bilo pomeranjem kursora, bilo uobičajenom šahovskom notacijom (ono: e2-e4 ili S16 itd.) Ako nam je volja, možemo okretati tablu da bismo je videli iz profila ili sa protivnikove strane. Naravno, kompjuter pamti sve poteze i može reprodukovati partiju, ili deo partije. Možemo usmiriti partiju u kojoj smo efektno pobedili, ili igrati sa živim partnerom umesto sa kompjuterom, ili rešavati šahovske probleme, ili reprodukovati tuđe partije...

... naravno, pod uslovom da nismo preveliki paceri, jer kompjuter je neumoran i nemilosrdan.

## TARZAN

Martech



Teško ćete ga prepoznati.

Nije to više onaj visoki atleta sličan trokrlinom šifonjeru: sad je prilično zdepast i ne deluje preterano spretno ali ipak je vičan tuči, skakanju, trčanju (i naravno!) uzletanju i spuštanju uz pomoć lijana.

To mu je ovde svaki čas potrebno, jer se opet našao na muci: izveani nevaljali poglavica kidnapovao je Džejn. Tarzan je mora naći u roku od tri dana, inače će biti zlo i naopako; kidnaperi se, po pravilu, ne šale.

Treba, dakle, stići do poglavice, a put je težak i opasan: lavovi, živi pesak, ruševni hramovi nabližakani nezgodnim iznadenjima, pećinski lavirint; i ljudi i prirodna kao da su se zaveriali protiv lorda Grejstoka.

Kad mu budete pomagali da se reši neprijatelja i oslobodi Džejn, budite precizni u asteriranju sa lijana, ne podcenjujte urođenike u tuči, i, ako stignete, crtajte mapu. Otupela je prostorna i gusta (a ni onaj lavirint nije naivan), imate velike šanse da se nepovratno izgubite.

## TRAPDOOR

Kapak  
Piranha



### Ludo i nezaboravno

Gore je soba, dole podrum. Na podu sobe: kapak. U podrumu: gotemo rutavo čudovište. U sobi: nedak zver. Zver voli klopnu više nego ista na svetu, i stalno nešto izvoljeva jednom čudovištu koje ima ulogu kuvara.

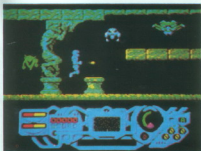
I u tome je drama. Čudovište mora, pre svega, da provali kako se zovu razne životinke kojima je okružen, da ne bi uhvatio pogrešnu i od nje spremio jelo. Ne sme propasti kroz kapak, ako mu je život mlo. Mora znati šta će sa drugim čudovištima koja svaki čas upadaju kroz kapak: jedna su korisna a druga opasna (ili prosto neupotrebljiva). Ne sme predugo držati kapak otvoren, i tako dalje: život mu prolazi između „mora“ i „ne sme“.

Donekle mu koriste lobanje: kad ih pokupi, daju mu dobre savete (posle čega mu se broj poena smanjuje). Koriste mu i neke osobine drugih čudovišta: jedno, na primer, bljuje vatru, i na to se vatri mogu zagrevati razna jela. (S druge strane, teško ga je posle oterati). Najkorisnije mu je da zaboravi logiku i osloni se na ludost: jer ovo je jedna sasvim otkaćena igra.

## Komodor

## SECRET ARMOUR OF ANTIPIAD

Tajni oklop antirijadski  
Palace Software



Dok je opasnost vrebala u svemiru, bilo je dene-dene; ali sad je baš dogorelo do nokata: vanzemaljci su došli i okupirali Zemlju. Zemljani civile i pište; jedinu šansu vide u tome da neko pronađe legendarni oklop, da se potom naoruža i opremi, i da najzad podmetne minu pod okupatorsku bazu.

Taj neko jeste visok i kršan momak, kakvi se nalaze samo u narodnim pesmama i na konkursima tipa „Mister Univerzum“. On zna šta mu je činiti, ali prvo mora da nađe oklop, što je samo po sebi dug i pogibeljan posao. Kad ga nađe i navuče, mora ići peške dok ne provali gde su antigravitacione cizme: onda može i leteti. Dalje su mu potrebni laseri, pa energija, pa odbrambeni mehanizam nazvan „poništivač čestica“, pa, na kraju, mina (Mina, promene radi, ne izaziva eksploziju nego implodiju).

Neprijatelja ima preko svake mere, sve jedan jeziviji od drugog; dogod ne nađe lasersko svetlosto oružje, momak ih može tući jedino kameonicama. Možete misliti kako mu je.

To jest, kako će mu biti; jer radnja se dešava u 2086. godini.

## GO FOR GOLD

Juriš na zlatnu  
Americana



Za većinu nas — ruku na srce — ovakve igre su jedini način da steknemo zlatnu (ili bilo kakvu) medalju.

Utehe, utehe nam treba.

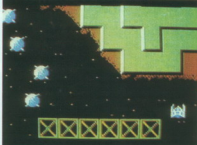
Ovde je možemo naći u šest disciplina: trkama na 100 metara i 110 metara s preponama, skokovima u vodu, skoku udaji, strelčarstvu i dizanju tereta.

Prethodno biramo garnirung: zemlju koju ćemo predstavljati, koje dresa itd. (možemo, čak, promeniti i naziv takmičenja pa napisati „Prvenstvo mladih komodorista“ ili „Memorijal vile Raviojeje“), ali najvažnijim mi se čini izbor super-nika. Ako smo sami, zna se ko će to biti; ako nismo, može učestvovati još petoro prijatelja (ili, naravno, marje).

Program je velikodušan: po želji daje i ponovljeni snimak. Ako baš hoćemo, možemo povrh toga i jurkati na svoj sopstveni rekord (u trkama). Tada je u stazi pored naše neke vrste našeg dvojnika: takmičar koji apsolutno neverno ponavlja našu rekordnu trku. Tako odmah vidimo na čemu smo... za razliku od stvarnog života, u kome je ta mogućnost nešto manja.

## WARHAWK

Ratni jastreb  
Firebird



Ima neke ložike u tome što je firma zvana „Žar-ptica“ proizvela igru sa pitčijim naslovom, ima u toj igri likovnih rešenja koja prijaju oku...

... ali to je uglavnom sve. Warhawk je pravljen po proverenoj formuli: letelica mašina, vertikalno skrolovani ekran, gomila svakojakih neprijatelja, pucanje, pucanje i opet pucanje.

Kako vreme prolazi, tako se formula obogaćuje u detaljima izvedbe i prezentaciji: ekrana je više, scenografija je masovitija, skrolovanje finije; osnovni princip, međutim, ostaje kakav je i bio.

Ovde ste pilot letelice iz 21. veka, misija vam je da uništite baze vanzemaljaca razmeštene po asteroidima i da obrorite što više neprijateljskih

letelica. (Što nije neka naročita novost: to ste bili već stotinama puta). Kad očistite jedan asteroid, krećete na drugi; kad očistite drugi krećete na treći...

... i tako dok vam ne dosadi.

## TRAILBLAZER

Krčiput  
Gremlin Graphics



Nemojte verovati naslovu. U ovoj igri nećete krčiti nikakve puteve.

Nije ih, uostalom, nešto naročito krčio ni programer. On je jamačno video da igre kao što su Spinidzyz, Kirel, Bobby Bearing, Marble Madness itd. itd. prolaze dobro na tržištu, ali nije htio da prepíše baš sve, pa je umesto krikera uzeo fudbalsku loptu...

Kola vi vodite niz neke svemirske drum. Drum je sav u raznobojnim kvadratima; neke su dve neutralne: siva i smeda. Bela je najbolja (i zato izuzetno retka), tamnoplava je dobra, zelena takođe; ostale izbegavajte, naročito crnu.

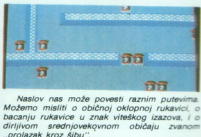
Vreme vam je ograničeno, broj skokova takođe (sedam odskoka po nivou). Ekran je podeljen na dva dela, pa će vam odmah biti jasno da je onaj drugi za suigrača (vašeg prijatelja ili vaš kompjuter). S prijateljem je lakše — ili bar manje teško.

Kao i uvek.

## GAUNTLET

Oklopna rukavica

U.S. Gold



Naslov nas može povesti raznim putevima. Možemo misliti o običnoj oklopnoj rukavici, o bacanju rukavice u znak viteškog izazova, ili o dirljivom srednjovekovnom običaju zvanom „prolazak kroz šibu“.

Ma šta misliti, bićemo u pravu. Ući ćemo u jednu od jako zamislivih varijanti davne prošlosti i naći na četvorčlanu ekipu: Popodijemo odmah da svaki član ima zaseban karakter i zasebne mogućnosti.

Budimo kavaljeri, počnimo od dame. Tira je svoju lepotu sakrila pod čvrst oklop, ali mač ne krije. Oboje joj treba kao hleb, jer veoma je sklona tući. Tor je ratnik kakvog nema, naročito u borbi prsa u prsa. Kvestor, vilenjak i strelac, lukar je kao tri lisice zajedno; njegova brzina priča je za sebe. Merlin je veštac (a šta bi drugo i bio s tim imenom?); onako stari i dostojanstven, nije rad tući, ali čini su mu prvorazredne.

Suštnja je u tome koji ćemo lik odabrati u kojoj situaciji. Sve ostalo je tuča do tuče i dvoboj do dvoboja sa karakondzulama svoje vrste, jer one nas ometaju u poteri za svanom, blagom, i čimim.

Ruku na srce, videli smo i boljih.

