



Mikrokomputery

Małgorzata Kalinowska-Iszkowska

Klucze

Wacław Iszkowski

DO BASICU

Amstrad

Apple

IBM PC

ZX Spectrum



# **Klucze do Basicu**

# Mikrokomputery

## Komitet Redakcyjny

*Sekretarz* WOJCIECH CELLARY

ZUZANNA GRZEJSZCZAK

ANDRZEJ KOBUS

*Przewodniczący* ROMUALD MARCZYŃSKI

PIOTR MISIUREWICZ

WOJCIECH NOWAKOWSKI

MACIEJ STOLARSKI

HALINA TEMPCZYK

JÓZEF WINKOWSKI

JAN ZABRODZKI

**Małgorzata Kalinowska-Iszkowska**  
**Wacław Iszkowski**

# **Klucze** **do Basicu**

Amstrad — Schneider  
Apple II  
IBM PC  
ZX Spectrum (Plus)



**Wydawnictwa Naukowo-Techniczne**  
**Warszawa 1987**

Opiniodawca *Magdalena Szeżyńska*  
Redaktor *Ewa Zdanowicz*  
Redaktor techniczny *Krzyszyna Orłoś*  
Okładkę projektował *Juliusz Rybicki*

681.3

W książce opisano najczęściej używane elementy poleceń języka Basic dla czterech mikrokomputerów: Amstrad — Schneider, Apple II, IBM PC i ZX Spectrum. Podano wykaz instrukcji języka oraz opis komunikatów błędów. Wyjaśniono też polskie znaczenie angielskich słów kluczowych języka. Książka jest przeznaczona dla szerokiego kręgu czytelników zajmujących się programowaniem mikrokomputerów.

© Copyright by Wydawnictwa Naukowo-Techniczne  
Warszawa 1987

All rights reserved  
Printed in Poland

ISBN 83-204-0809-1

# Spis treści

Przedmowa/7

1. Elementy poleceń/9

2. Amstrad — Schneider/16

3. Apple II/33

4. IBM PC/51

5. ZX Spectrum (Plus)/71

6. Wykaz słów kluczowych języka/84

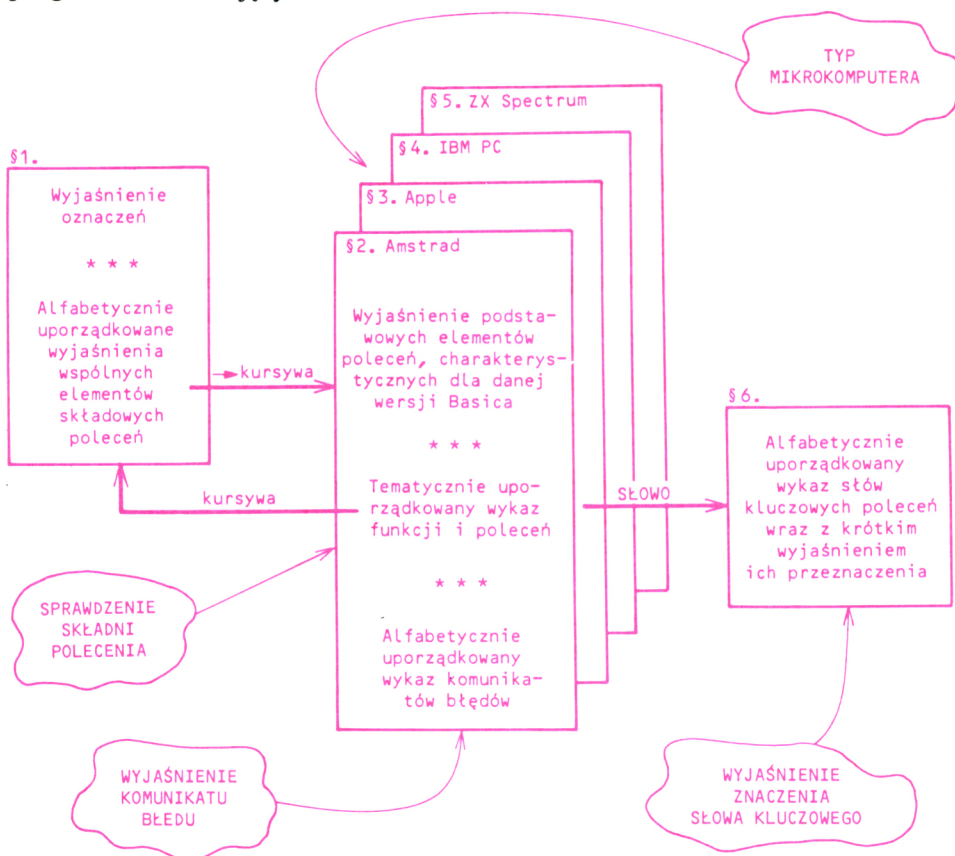
Literatura/92





# Przedmowa

Przeznaczeniem tej książeczki jest szybka pomoc w sprawdzeniu postaci lub interpretacji polecenia zapisanego w języku Basic dla jednego z czterech typów mikrokomputerów. Mówiąc językiem szkolnym, jest to bryk, czyli ściągawka, a więc może dobrze służyć tylko tym, którzy już coś niecoś wiedzą o programowaniu w języku Basic.



Bryk ten obejmuje jedynie najważniejsze i najczęściej używane elementy języka przy uwzględnieniu zasad wprowadzania tekstu programu i efektywnego wykorzystania wszystkich funkcji klawiatury mikrokomputera. Pominęto te elementy języka, których zastosowanie wymaga już znacznej wiedzy i jest ściśle związane z cechami funkcjonalnymi sprzętu. Sposób korzystania ze ściągawki przedstawia rysunek.

Dla rozróżnienia — *poleceniem* będziemy dalej nazywali każdą wprowadzoną linię przeznaczoną dla interpretatora systemu Basic. Linia rozpoczynająca się od słowa kluczowego jest *zleceniem* do natychmiastowego wykonania w trybie bezpośrednim. Linia rozpoczynająca się od numeru linii jest *instrukcją* lub *sekwencją instrukcji*, które są odkładane w pamięci operacyjnej jako fragment programu wykonywanego dopiero po wprowadzeniu zlecenia RUN.

Książka ta umożliwia więc sprawdzenie składni poleceń języka, zorientowanie się w możliwościach języka oraz wskazuje kierunki poszukiwań nowych konstrukcji dla realizacji programu. Ujednolicanie opisu różnych wersji języka daje też możliwość łatwego ich porównania, a więc na przykład ułatwia przeniesienie istniejącego programu na inny typ komputera.

W opisie języka zamieszczono też w porządku alfabetycznym wykaz komunikatów błędów, które mogą być wyprowadzane przez system Basic. Wykaz ten uzupełniono numerami błędów, które są używane przy programowej obsłudze sytuacji wyjątkowych. Ostatni rozdział dla ułatwienia zrozumienia treści programów wyjaśnia znaczenia angielskich słów kluczowych języka.

*Autorzy*

Warszawa, w październiku 1986

# 1. Elementy poleceń

## Oznaczenia

W wykazie poleceń dla każdej wersji języka Basic zastosowano następujące konwencje zapisu składni:

### DUŻE LITERY

Słowo kluczowe identyfikujące polecenie lub jego fragment. Znaczenie słów kluczowych podano w rozdz. 6.

### *kursywa*

Element składni polecenia reprezentujący opisowo jego parametr. Wyjaśnienie poszczególnych terminów identycznych dla kilku wersji języka podano poniżej. Pozostałe terminy wyjaśniono w rozdziałach dotyczących konkretnej wersji języka Basic (wskazane są one przez → *termin*).

### [ ]

Nawiasy kwadratowe oznaczają, że zawarty w nich fragment polecenia może występować opcjonalnie.

...

Wielokropek oznacza, że poprzedzający go element składni, lub fragment ujęty w nawiasy kwadratowe, może być wielokrotnie powtórzony.

### KLAW\_\_1

Oznaczenie klawisza z klawiatury — np. CTRL, ENTER.

### KLAW\_\_1 KLAW\_\_2

Prezentacja sekwencji dwóch kolejno wciskanych klawiszy.

### KLAW\_\_1 & KLAW\_\_2

Prezentacja zestawu dwóch klawiszy wciskanych równocześnie.



Oznaczenie funkcji standardowych w wykazie poleceń.

*adres*

Wartość całkowita podająca adres bajtu w pamięci operacyjnej — zawarta w zakresie dostępnej przestrzeni adresowej.

*arg*

Wyrażenie typu zgodnego z typem parametru funkcji standardowej lub definiowanej — podawane przy wywołaniu funkcji.

*cyfra<sub>p</sub>*

Jedna z cyfr dopuszczalnych w systemie zapisu o podstawie *p*:

2 — 0 1

10 — 0 1 2 3 4 5 6 7 8 9

8 — 0 1 2 3 4 5 6 7    16 — 0 1 2 3 4 5 6 7 8 9 A B C D E F

*dłg<sub>tekstu</sub>*

Wyrażenie całkowite, którego wartość określa liczbę znaków w → *wartości tekstowej*.

*dowolny<sub>tekst</sub>*

Komentarz będący dowolnym tekstem pomijanym przez translator podczas tłumaczenia treści polecenia.

*inkrement*

Liczba naturalna podająca różnicę między kolejno automatycznie generowanymi → *numerami linii* (przeważnie równa 10).

*instrukcja*

Polecenie przeznaczone do odłożenia w pamięci operacyjnej komputera jako fragment programu. Wykonanie instrukcji następuje po podaniu zlecenia RUN. Dla określenia kolejności instrukcji w programie, są one w → *linii programu* poprzedzane numerem linii.

*instrukcje*

Sekwencja instrukcji, znajdujących się w tej samej → *linii programu*, oddzielonych od siebie znakiem dwukropka.

*kąt*

Wartość w radianach określająca część lub początek i koniec łuku okręgu, który ma być rysowany.

*kod<sub>znaku</sub>*

Wartość całkowita stałej lub wyrażenia z zakresu 0..255 reprezentująca kod znaku według przyjętego standardu kodów ASCII(ISO).

*kod<sub>ster</sub>*

Wyrażenie całkowite przypisujące wartość kodu znaku danego klawisza → *klawiatURY* w zależności od wykorzystania klawiszy sterujących SHIFT i CTRL.

*liczba<sub>spacji</sub>*

Wartość całkowita dodatnia będąca argumentem funkcji przesuwej kursor w prawo z wydrukowaniem lub wyświetleniem podanej liczby spacji (odstępów).

*nazwa\_\_fun*

*nazwa\_\_fun\_\_aryt      nazwa\_\_fun\_\_tekst*

Identyfikator poprzedzony standardowym identyfikatorem FN i uzupełniony znacznikiem określającym → *typ funkcji*.

*nazwa\_\_tabl*

Identyfikator tablicy uzupełniony znacznikiem określającym → *typ zmiennej*.

*nr\_\_atramentu*

Wartość wyrażenia całkowitego będąca numerem atramentu wiązany z kolorem z → *palety barw*, którym na ekranie jest rysowany kształt znaku (w trybie → *ekranu tekstowego*) lub punkt rysunku (w trybie → *ekranu graficznego*).

*nr\_\_błędu*

Wartość całkowita z zakresu 0..255 identyfikująca błąd, jaki ma być wykryty podczas wykonania instrukcji ERROR.

*nr\_\_klawisza*

Wartość całkowita wskazująca jeden z umownie ponumerowanych klawiszy → *klawiatury*.

*nr\_\_kol*

Numer kolumny dla pozycji znaku w wierszu wyświetlanym na ekranie lub drukowanym. Dodatnia wartość całkowita musi się zawierać w zakresie → *szerokości wiersza* określonej standardowo organizacją → *ekranu tekstowego* lub deklarowaną. Wartość spoza zakresu jest błędna, bądź jest liczona modulo szerokość wiersza.

*nr\_\_koloru*

*nr\_\_koloru\_\_ramki      nr\_\_koloru\_\_tła*

Wyrażenie całkowite wskazujące na jeden z kolorów z → *palety barw*. Kolorem tym zostanie wypełniona przestrzeń ramki ekranu, tła znaku lub tła rysunku — tzw. papieru.

*nr\_\_linii*

Liczba naturalna należąca do zakresu dopuszczalnej numeracji → *linii programu*.

*nr\_\_linii\_\_lub\_\_instrukcje*

Alternatywna możliwość umieszczenia w poleceniu numeru linii lub sekwencji instrukcji. W pierwszym przypadku zapis *nr\_\_linii* jest równoważny zapisowi GOTO *nr\_\_linii*.

*nr\_\_linii\_\_nowy      nr\_\_linii\_\_stary*

Stary numer linii wskazuje pierwszą linię, od której ma być dokonane przenumerowanie linii zaczynając od podanego nowego numeru linii z podanym też inkrementem (standardowo starym numerem linii jest pierwsza linia programu, a nowym — numer 10).

*nr\_\_portu*

Wartość wyrażenia całkowitego, przeważnie z zakresu 0..255, wskazująca jeden z portów dla operacji wejścia-wyjścia.

*nr\_\_pozycji*

Wartość wyrażenia całkowitego wskazująca położenie znaku w  $\rightarrow$  *wartości tekstowej* będącej stałą, wyrażeniem lub zmienną tekstową.

*nr\_\_str*

Wartość całkowita określająca  $\rightarrow$  *numer logiczny strumienia* wskazujący źródło wprowadzanych danych lub przeznaczenie wyprowadzanych wartości. Strumieniem jest tutaj ciąg wartości reprezentowanych przez sekwencję znaków.

*nr\_\_wiersza*

Wartość całkowita z zakresu numeracji wierszy określonej organizacją  $\rightarrow$  *ekranu tekstowego* wskazująca wiersz, w którym ma być ustawiony kursor wyświetlania kolejnego znaku.

*okres*

Wyrażenie całkowite o wartości określającej liczbę jednostek czasu odliczanych przez zegar czasu rzeczywistego działający w systemie.

*par*

Parametr funkcji definiowanej będący dowolnym dopuszczalnym w języku identyfikatorem, który umieszczony w wyrażeniu definicji funkcji wskazuje miejsce podstawienia argumentów.

*parXYZ parA..Z*

Stała całkowita będąca parametrem polecenia o jednej z możliwych podanych wartości *X*, *Y* lub *Z*, bądź z zakresu *A*, *B*,...*Z*.

*polecenie*

Treść pojedynczej  $\rightarrow$  *linii programu*, będącej zleceniem lub instrukcją, wprowadzonej zgodnie ze składnią linii polecenia.

*promień*

Wyrażenie całkowite o wartości deklarującej w liczbie punktów promień rysowanego na  $\rightarrow$  *ekranie graficznym* okręgu.

*przes\_\_x przes\_\_y*

Wyrażenia całkowite o wartościach deklarujących w liczbie punktów widoczne lub niewidoczne przesunięcie kursora na  $\rightarrow$  *ekranie graficznym* w poziomie *x* i w pionie *y*, względem jego aktualnej pozycji.

*rozm*

Stała lub wyrażenie całkowite o wartościach deklarujących liczbę elementów w danym wymiarze tablicy ( $\rightarrow$  *deklaracja tablicy*). Z zasady należy doliczyć element o wskaźniku równym 0, jeżeli nie wykonano wcześniej instrukcji OPTION BASE 1.

*rozmiary*

Lista rozmiarów dla każdego wymiaru → *deklaracji tablicy*.

*sep*

Separator będący przecinkiem lub średnikiem, sterujący rozmieszczanie wyprowadzanych w wierszu wartości. Przecinek przesuwa kursor wyprowadzania na początek kolejnej strefy określonej → *szerokością strefy*. Średnik skleja kolejno wyprowadzane wartości według wzorca standardowego.

*składnia linii polecenia*

Linia polecenia w języku Basic posiada następującą składnię:

[*nr\_linii*] *instrukcja* [*:instrukcja*]...[*:komentarz*] ENTER

Polecenie nie zawierające *nr\_linii* jest zleceniem. Końcem linii jest znak sterujący wprowadzany klawiszem ENTER (lub RETURN).

*spec\_pliku*

Wyrażenie tekstowe → *identyfikacji pliku* zgodnie z zasadami określonymi dla danego systemu.

*stała*

*stała\_aryt*    *stała\_calk*    *stała\_tekst*

Bezpośrednio zapisana → *wartość liczbowa* lub → *wartość tekstowa*.

*tekst\_zachęty*

Tekst złożony z dowolnych znaków, wyświetlany podczas wykonywania instrukcji wprowadzania INPUT.

*typ\_pliku*

Pojedyncza litera specyfikująca → *postać zapisu pliku* na nośniku.

*wart\_pocz*    *wart\_końc*    *wart\_kroku*

Wyrażenia arytmetyczne o wartościach określających parametry wykonywania instrukcji cyklu wyliczanego.

*wprowadzanie tekstu programu*

Tekst programu wprowadzamy pojedynczymi → *liniami programu* zgodnie ze składnią linii polecenia. W pamięci tekst programu jest porządkowany według rosnącej numeracji linii. W trakcie wprowadzania linii można skorzystać z możliwości → *poprawiania linii programu* zanim zakończymy ją znakiem klawisza ENTER (lub RETURN). Cały tekst programu można wielokrotnie poprawiać usuwając pojedyncze linie przez wprowadzenie tylko ich numeru linii, bądź zleceniem DELETE. Wprowadzenie linii o istniejącym już w tekście numerze linii zastępuje dotychczasową jej treść nowym tekstem. Możliwe jest też częściowe → *poprawianie linii programu*. Do wyświetlenia aktualnej treści tekstu programu służy zlecenie listowania LIST.

*wsp*

*wsp\_\_x*      *wsp\_\_x1*      *wsp\_\_x2*

*wsp\_\_y*      *wsp\_\_y1*      *wsp\_\_y2*

Wyrażenia całkowite o wartościach określających w liczbie punktów nowe położenie kursora w poziomie *x* i w pionie *y* na →*ekranie graficznym*.

*wyrazenie*

Dowolna formuła poprawnie skonstruowana z dopuszczalnych →*operatorów*, nawiasów okrągłych i argumentów. Argumentem może być stała, zmienna lub wywołanie funkcji standardowej i definiowanej — jednego z dopuszczalnych →*typów*. Pojedynczy argument jest też wyrażeniem.

*wyr\_\_adr*

Wyrażenie całkowite o wartości podającej adres bajtu w pamięci operacyjnej komputera z zakresu dostępnej przestrzeni adresowej.

*wyr\_\_aryt*

Wyrażenie skonstruowane z →*operatorów* arytmetycznych i argumentów →*typu* liczbowego (całkowitego lub rzeczywistego).

*wyr\_\_calk*

Wyrażenie arytmetyczne, którego ostateczna wartość zostaje →*zao-krąglona* do wartości całkowitej.

*wyr\_\_log*

Wyrażenie logiczne o →*wartości logicznej* wykorzystujące →*operatory* relacji oraz logiczne.

*wyr\_\_nr\_\_linii*

Wyrażenie o →*wartości liczbowej* (całkowitej) — będącej numerem linii istniejącym w programie.

*wyr\_\_tekst*

Wyrażenie tekstowe będące stałą, zmienną lub funkcją (→*typu tekstowego*) — ewentualnie złożoną z następnym argumentem tego typu poprzez →*operator* tekstowy konkatencji.

*wzorzec*

Wyrażenie tekstowe opisujące symbolicznie lokalną zasadę formatowania wyprowadzanych w danym poleceniu →*wartości* →*typu* liczbowego oraz tekstowego.

*zakres\_\_linii*

Określenie w postaci →*opisu zakresu numeracji linii* fragmentu tekstu programu poddanego działaniu polecenia.

*zakres\_\_liter*

Definicja →*typu zmiennej lub funkcji* może być dokonana poprzez



specyfikację typu pierwszą literą → *identyfikatora*. Zakres liter przypisanych typowi jest określany przez listę pojedynczych *liter* lub formuł typu *litera\_\_1 - litera\_\_2* obejmując wszystkie litery w tym zakresie.

*zlecenie*

Polecenie przeznaczone do natychmiastowego wykonania. W stosunku do instrukcji odróżnia się brakiem numeru linii.

*zmienna*

Obiekt programu przechowujący → *wartość* o danym → *identyfikatorze* i → *typie*.

*zm\_\_aryt*

Zmienna prosta lub wskaźnikowa przechowująca → *wartość liczbowa*.

*zm\_\_ster*

Zmienna → *typu* liczbowego przechowująca licznik dla instrukcji cyklu wyliczanego.

*zm\_\_tekst*

Zmienna prosta lub wskaźnikowa przechowująca → *wartość tekstową* lub kod pojedynczego znaku.

## 2. Amstrad—Schneider

Mikrokomputer osobisty Amstrad CPC 464, i równoważny mu Schneider, jest standardowo wyposażony w interpretator języka Amstrad BASIC firmy Locomotive Software's Mallard. Program interpretatora języka jest umieszczony w osobnej pamięci stałej ROM. Ta wersja języka Basic jest równoważna wersji BASIC Microsoft z bogatym uzupełnieniem w zakresie grafiki i generacji dźwięku oraz obsługi asynchronicznych zdarzeń.

Omówiona tutaj wersja języka jest ograniczona do konfiguracji komputera wyposażonego tylko w pamięć na kasetach. Bogatsze wersje mogą być wyposażone w adaptory dyskietek 3-calowych.

### *deklaracja tablic*

Tablice mogą mieć dowolną liczbę wymiarów i rozmiarów (w zakresie liczb całkowitych) z minimalną wartością wskaźnika równą 0. Tablice są deklarowane dynamicznie w trakcie wykonywania programu i po wymazaniu mogą być powtórnie zadeklarowane. Istnieje również deklaracja standardowa tablicy według pierwszej użytej zmiennej wskaźnikowej.

### *identyfikator*

Identyfikatorem jest ciąg do 40 znaków — złożony z liter, cyfr i kropki — rozpoczynający się od litery. Duże i małe litery są utożsamiane. Standardowo identyfikatory są listowane zawsze z wykorzystaniem małych liter. Identyfikatory nie mogą być →*słowami kluczowymi*. Identyczne identyfikatory (z rozróżnieniem znacznikiem typu) mogą być stosowane dla różnych →*typów zmiennych i funkcji*.

### *ekran graficzny*

Ekran monitora graficznego jest utożsamiony z →*ekranem tekstowym* i może funkcjonować w jednym z trzech trybów: niskiej, zwykłej i wysokiej rozdzielczości wybieranym poleceniem:

MODE 0 — 160 × 200 punktów w jednym z 16 atramentów każdy,  
 MODE 1 — 320 × 200 punktów w jednym z 4 atramentów każdy,  
 MODE 2 — 640 × 200 punktów w jednym z 2 atramentów każdy.

Zakres zmian współrzędnych ekranu dla każdego trybu jest określony w poziomie  $wsp\_x$ : 0.639 oraz w pionie  $wsp\_y$ : 0.399 ze standardowym położeniem punktu początkowego układu w lewym dolnym rogu ekranu. Położenie początku układu może być redefiniowane. Możliwość podziału ekranu na okna jest identyczna jak dla ekranu tekstowego. Kolory atramentu są wybierane z  $\rightarrow$  *palety barw*.

### *ekran tekstowy*

Ekran tekstowy monitora (monochromatycznego lub kolorowego) jest utożsamiany z  $\rightarrow$  *ekranem graficznym* i może działać w tych samych trzech trybach umożliwiając wyświetlenie:

MODE 0 — 25 wierszy po 20 znaków w 16 kolorach,  
 MODE 1 — 25 wierszy po 40 znaków w 4 kolorach,  
 MODE 2 — 25 wierszy po 80 znaków w 2 kolorach.

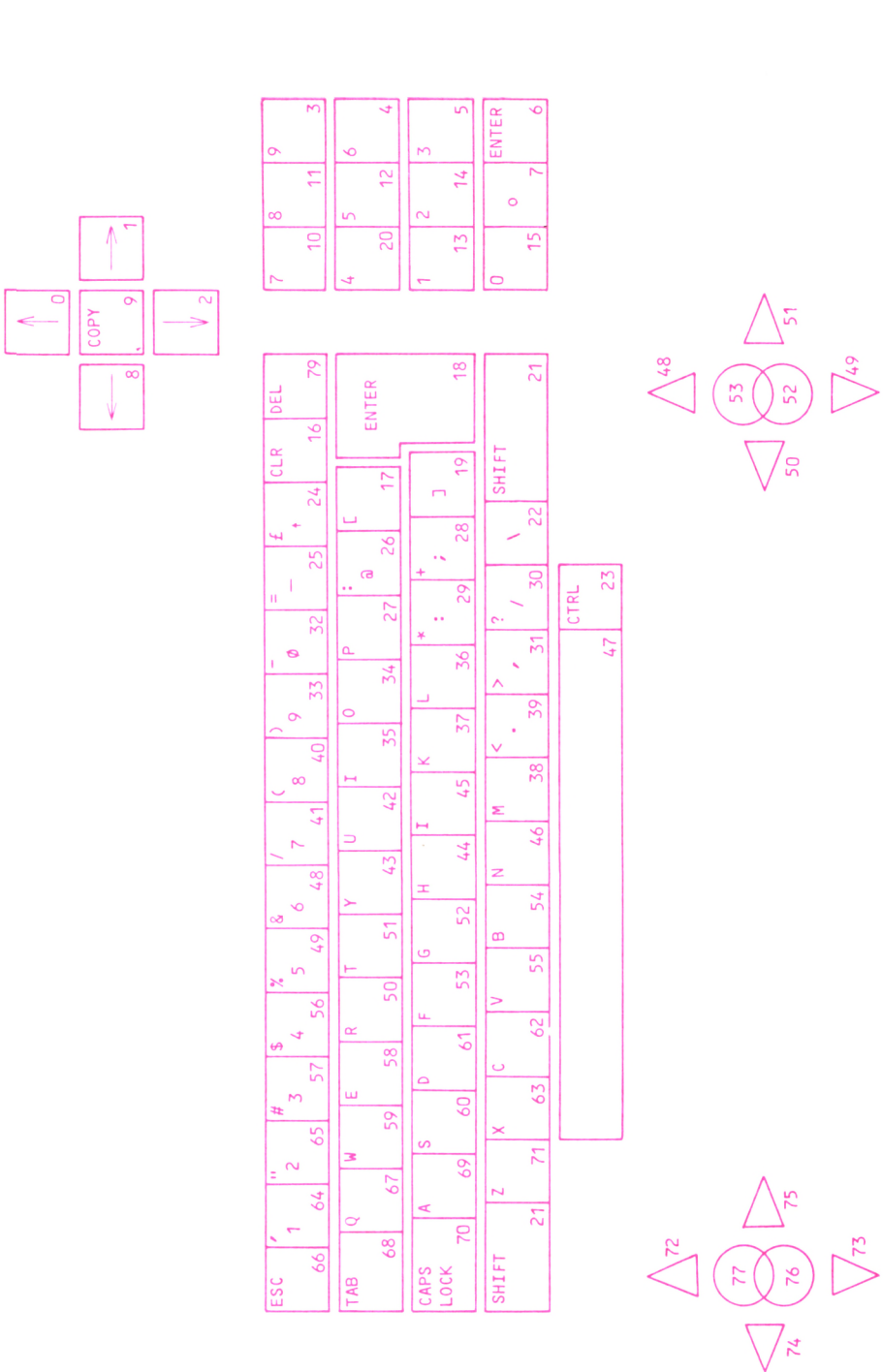
Zakres numeracji wierszy jest określony w przedziale 1..25, a kolumn w przedziale 1..255 (modulo liczba znaków w wierszu). Pozycja początkowa kursora wyświetlania jest określona w lewym górnym rogu ekranu. Do programowego przesuwania kursora w określone miejsce ekranu służy instrukcja LOCATE oraz separatory instrukcji wprowadzania. Ekran tekstowy (również wtedy ekran graficzny) można logicznie podzielić na okna, uzyskując możliwość wyprowadzania jednocześnie do 8 różnych strumieni (określonych  $\rightarrow$  *numerem logicznym strumienia*). Nie wszystkie okna mogą być w danej chwili wyświetlane.

### *identyfikacja pliku*

Wyrażenie tekstowe zawierające dowolne znaki z wyjątkiem ! na początku tekstu. Wykrzyknik wstrzymuje wyświetlanie informacji o przebiegu wykonywania zlecenia ładowania programu.

### *klawiatura*

Klawiatura jest ułożona w standardowym układzie QWERTY klawiszy wprowadzających kody znaków: małych i dużych liter, cyfr, symboli oraz znaków sterujących. Dodatkowo istnieje wyróżniony fragment klawiatury funkcyjno-numerycznej oraz przesuwania kursora. Wszystkie klawisze na klawiaturze oraz klawisze dwóch manipulatorów mają przypisane odrębne numery z zakresu 0..79 (rys. 2.1). Korzystając z polecenia KEYDEF możliwe jest przypisanie każdemu klawiszowi o podanym numerze (z wyjątkiem ESC SHIFT CTRL) nowej wartości kodu znaku.



Rys. 2.1

*klawisze funkcyjne*

Istnieje możliwość zdefiniowania 32 klawiszy funkcyjnych o zakresie numeracji 0..31, którym są przypisane odpowiednie kody znaków z zakresu 128..159. Każdemu klawiszowi funkcyjnemu może być przypisany dowolny tekst, wprowadzany w całości po jednorazowym naciśnięciu klawisza. Suma długości wszystkich tekstów nie może przekraczać 120 znaków. Standardowo są zdefiniowane klawisze z klawiatury funkcyjno-numerycznej (o kodach z zakresu 128..139) bez przypisania im tekstów oraz klawisze CTRL&ENTER (o kodzie 140) z przypisanym tekstem RUN". Korzystając z możliwości zmiany kodu znaku dla każdego ponumerowanego klawisza, możliwe jest przypisanie kodu znaku funkcyjnego klawiszowi, a następnie przypisanie jemu dowolnej funkcji w postaci tekstu.

*klawisze sterujące*

Wyróżnione są następujące klawisze o kodach znaków sterujących:

- |           |   |
|-----------|---|
| ENTER     | — zakończenie wprowadzania lub poprawiania linii zawierającej polecenie dla systemu Basic.  |
| SHIFT     | — wprowadzanie dużej litery lub symbolu wskazanego równocześnie wciśniętym klawiszem.   |
| CAPS_LOCK | — przełączenie na stałe trybu wprowadzania liter tylko jako dużych — odwołanie przełączenia powtórным wciśnięciem klawisza CAPS_LOCK. |
| CTRL      | — wprowadzanie znaku sterującego związanego z równocześnie wciśniętym klawiszem.  |
| ESC       | — przerwanie wykonywania zlecenia, również programu; usunięcie aktualnie wprowadzanej linii.  |
| DEL       | — usunięcie znaku poprzedzającego kursor wprowadzania z przesunięciem pozostałej części tekstu w lewo.                                |
| CLR       | — usunięcie znaku wskazanego kursorem wprowadzania z przesunięciem pozostałej części tekstu w lewo.                                   |
| TAB       | — przesunięcie kursora w prawo do najbliższej pozycji tabulacji n*8.  |
| COPY      | — skopiowanie znaku wskazanego kursorem kopiowania w miejsce wskazane kursorem wprowadzania.  |
|           | — przesuwanie kursora wprowadzania edycji odpowiednio:  |
| ↑         | — w górę,   |
| ←         | — w lewo,   |
| →         | — w prawo,  |
| ↓         | — w dół   |
- w ramach aktualnie wprowadzanej linii lub po całym ekranie, jeżeli po ostatnim ENTER nie wprowadzono jeszcze żadnego znaku.

Pozostałe funkcje sterujące uzyskuje się przez jednoczesne działanie klawiszy sterujących z innymi klawiszami:

- ESC ESC — przerwanie wykonywania polecenia RUN i zakończenie wykonywania programu.
- ESC SPACE — przerwanie wykonywania programu i następnie jego wznowienie.
- ESC *znak* — przerwanie wykonywania programu i następnie jego wznowienie z zapamiętaniem w buforze kodu wprowadzonego znaku.
- ESC&SHIFT&CTRL — wyzerowanie stanu komputera.
- CTRL&SHIFT — przełączenie na stałe działanie klawisza SHIFT, aż do powtórnego wprowadzenia CTRL&SHIFT.
- CTRL&ENTER — wprowadzenie tekstu RUN".
- przesuwanie kursora kopiowania po ekranie:
- SHIFT&↑ — w górę,
- SHIFT&← — w lewo,
- SHIFT&→ — w prawo,
- SHIFT&↓ — w dół.
- przesuwanie kursora wprowadzania edycji po ekranie na:
- CTRL&↑ — początek bieżącej linii.
- CTRL&← — początek bieżącego wiersza.
- CTRL&→ — koniec bieżącego wiersza.
- CTRL&↓ — koniec bieżącej linii.
- CTRL&TAB — przełączenie trybu wprowadzania na wstawianie następnie wprowadzanych znaków w tekst w linii zgodnie ze wskazaniem kursora i z przesuwaniami pozostałej prawej części tekstu w prawo; wyłączenie tego trybu następuje po powtórnym wprowadzeniu CTRL&TAB.

### *lewy, prawy, góra, dół*

Wyrażenia całkowite o wartościach wyznaczających granice okna na →*ekranie graficznym* w zakresie zmienności współrzędnych →*wsp\_x*, *wsp\_y*, lub na →*ekranie testowym* w zakresie liczby wierszy i kolumn.

### *linia programu*

W linii może być do 255 znaków. Zakres numeracji linii wynosi 1..65535.

*numer logiczny strumienia*

Wyrażenie całkowite o wartości z zakresu 0..9 wskazujące numer strumienia:

- danych wprowadzanych:
  - 0..7 z klawiatury z wyświetlaniem kopii tekstu wprowadzanego w odpowiednim oknie ekranu,
  - 8 z klawiatury z wyprowadzaniem kopii tekstu na drukarkę,
  - 9 z pliku wejściowego z pamięci kasetowej,
- wyprowadzanych wyników:
  - 0..7 do odpowiedniego okna ekranu,
  - 8 na drukarkę,
  - 9 do pliku wyjściowego w pamięci kasetowej.

*numer stopera*

Wyrażenie całkowite o wartości z zakresu 0..3 wskazującej jeden ze stoperów odliczających czas rzeczywisty.

*operatory*

- arytmetyczne:  $\uparrow$  - \* / \ MOD + -
- relacji : < <= =< = >= ==> > <>
- logiczne : NOT AND OR XOR
- tekstowe : + oraz relacji

Operatory logiczne funkcjonują też na rozwinięciach dwójkowych wartości liczbowych całkowitych.

*opis zakresu numeracji linii*

Zakres numeracji linii określa fragment programu, który ma być poddany działaniu zlecenia i może przyjmować postać:

- pusty — cały tekst programu,
- nr\_linii* — tylko linia o podanym numerze,
- nr\_linii* - — fragment od danej linii do końca tekstu programu,
- *nr\_linii* — fragment od pierwszej linii programu do podanej włącznie,
- nr\_linii* - *nr\_linii* — fragment od pierwszej podanej linii do drugiej włącznie.

*opis znaku*

Każdy znak o kodzie z zakresu 0..255 może mieć zdefiniowany nowy kształt przez podanie wartości 8 kolejnych bajtów jego rastra jako

rozwińnięcie wartości w postać dwójkową z zaczernionymi polami wskazanymi wartością równą 1. Standardowo dopuszczalne jest redefiniowanie znaków o kodach z zakresu 240..255.

### *opóźnienie*

Wyrażenie całkowite o wartości określającej w liczbie jednostek czasu okres wstrzymania działania programu.

### *paleta barw*

Dla monitora kolorowego jest dostępnych 27 kolorów:

0 czarny	9 zielony	18 jasnozielony
1 niebieski	10 siny	19 morski
2 jasnoniebieski	11 błękitny	20 jasnosiny
3 czerwony	12 żółty	21 wapienny
4 karminowy	13 biały	22 pastelowozielony
5 bladopurpurowy	14 pastelowoniebieski	23 pastelowosiny
6 jasnoczerwony	15 pomarańczowy	24 jasnożółty
7 purpurowy	16 różowy	25 pastelowożółty
8 jasnokarminowy	17 pastelowokarminowy	26 jasnobiały

Numery kolorów są wykorzystywane w definicji koloru dla każdego z 16 atramentów, koloru papieru (tła) oraz ramki.

### *parametry\_\_opisu\_\_melodii*

Lista parametrów opisujących w każdym z kanałów częstotliwość, czas trwania, amplitudę, itp. wartości precyzujące sposób generowania dźwięków.

### *poprawianie linii programu*

Pojedyncze linie programu w trakcie wprowadzania lub sprowadzone poleceniem EDIT lub z wykrytym podczas wykonywania błędem składniowym (*Syntax Error*) mogą być poprawiane następująco:

- przesunąć kursor wprowadzania w błędne miejsce linii — korzystając z klawiszy sterujących ↑ ← → ↓,
- usunąć znaki z treści linii klawiszami DEL lub CLR,
- wymienić błędne znaki na nowe w treści linii, wprowadzając je w miejsce wskazane kursorem,
- wpisać dodatkowe nowe znaki w treść linii w miejsce wskazane kursorem po przełączeniu CTRL&TAB aż do ponownego użycia CTRL&TAB,
- ewentualnie skopiować tekst istniejący na ekranie (np. wylistowany poleceniem LIST), przesuwać na początek żądanego fragmentu tekstu kursor kopiowania klawiszami SHIFT&↑, SHIFT&←,



SHIFT&→, SHIFT&↓, a następnie klawiszem COPY kopiować kolejne znaki w miejsca wskazane kursorem wprowadzania w poprawianej linii,

— zakończyć poprawianie linii klawiszem ENTER.

#### *postać zapisu pliku*

W pamięci kasetowej mogą być zapisywane pliki następujących typów:

brak \$ — plik z programem w Basicu,

P % — plik z programem w Basicu z protekcją odczytu,

A \* — plik w kodzie ASCII,

B & — plik w postaci binarnej,

' — plik w postaci binarnej z protekcją odczytu.

Litera P, A, B lub jej brak określa odpowiednio typ pliku przy jego składowaniu, a symbole \$ % \* & ' identyfikują typ pliku podczas wyświetlania listy plików znajdujących się na kasecie.

#### *rodzaj pliku*

Przy współpracy z pamięcią kasetową możliwe jest otwarcie dwóch rodzajów plików:

OPENIN — plik wejściowy dla odczytu,

OPENOUT — plik wyjściowy dla zapisu w blokach o długości rekordu 24 bajty.

#### *słowo kluczowe*

Język zawiera 160 słów kluczowych, które są zastrzeżone i nie mogą być używane jako → *identyfikatory* zmiennych. W trakcie listowania tekstu programu słowa kluczowe są zawsze wyprowadzane dużymi literami.

#### *szer\_\_strefy*

Standardowa szerokość strefy na wydruku i na ekranie wynosi 13 znaków. Może być ona zmieniana w zakresie 1..255 znaków.

#### *szer\_\_wiersza*

Standardowa szerokość wiersza na wydruku wynosi 132 znaki. Może być ona zmieniana w zakresie 1..255 znaków. Szerokość wiersza równa 255 oznacza dla drukarki wyprowadzenie linii bez uzupełniania jej znakiem zmiany wiersza.

#### *typ zmiennej i funkcji*

Istnieją trzy typy zmiennych i funkcji deklarowanych pierwszą literą identyfikatora określoną zleceniem DEFxxx, lub też przez uzupełnienie identyfikatora znacznikiem typu

% — liczbowego całkowitego,

! — liczbowego rzeczywistego,

\$ — tekstowego.

Brak znacznika typu jest równoważny typowi liczbowemu rzeczywistemu.

#### wartość liczbowa

Dopuszczalne są wartości liczbowe typu:

— całkowitego z zakresu  $-32768..32767$  zapisane w postaci:

cyfra\_\_10 ... dziesiętnej,

&cyfra\_\_16 ... szesnastkowej,

&Hcyfra\_\_16 ... szesnastkowej,

&Xcyfra\_\_2 ... dwójkowej.

(W niektórych zastosowaniach wartość całkowita jest traktowana jako ciąg 16 bitów w zapisie uzupełnień do dwóch.)

— rzeczywistego z zakresu

$-1,7 \cdot 10^{38} .. -2,9 \cdot 10^{-39}$ ,  $0$ ,  $2,9 \cdot 10^{-39} .. 1,7 \cdot 10^{38}$

z dokładnością do 9 cyfr znaczących w postaci liczby całkowitej spoza zakresu liczb całkowitych, ułamkowej i wykładniczej.

#### wartość logiczna

Wartość logiczna prawda ( $-1$ ), fałsz ( $0$ ). Operatory logiczne działają na pełnym rozwinięciu 16 bitów  $\rightarrow$  wartości liczbowej całkowitej.

#### wartość tekstowa

Teksty mogą zawierać od 0 do 255 znaków o dowolnej wartości kodu znaku z zakresu 1..255.

#### wywołanie systemu Basic

Po włączeniu mikrokomputera oraz po każdym jego zerowaniu klawiszami CTRL&SHIFT&ESC, oraz po wydaniu zlecenia NEW zgłasza się system Basic wyprowadzając na ekran tekst:

Amstrad 64K Microcomputer(V1)

© 1984 Amstrad Consumer Electronics plc.

and Locomotive Software Ltd.

BASIC 1.0

Ready



gdzie Ready jest znakiem zachęty na wprowadzanie kolejnej linii programu (polecenia).

#### wyr\_\_calk\_\_lub\_\_<sup>(a)</sup> zmienna

Element listy argumentów wywołania podprogramu zewnętrznego będący wyrażeniem całkowitym lub adresem zmiennej podanym jako <sup>(a)</sup> zmienna.

#### zaokrąglenie

Zaokrąglenie wartości rzeczywistej do najbliższej wartości całkowitej.

**Wykaz poleceń***Edycja programu*

NEW  
 AUTO [*nr\_linii*] [,*inkrement*]  
 <ESC> 'wstrzymanie AUTO  
 LIST [*zakres\_linii*] [,*#nr-str*]  
 DELETE *zakres\_linii*  
 EDIT *nr\_linii*  
 RENUM [*nr\_linii\_nowy*] [, [*nr\_linii\_stary*] [,*inkrement*]]

*Komentarze*

REM *dowolny\_tekst*  
 '*dowolny\_tekst*

*Uruchomienie i śledzenie programu*

RUN [*nr\_linii*]  
 RUN *spec\_pliku*  
 CONT  
 TRON  
 TROFF

*Ladowanie, składowanie i wiązanie programów*

SAVE *spec\_pliku* [,*typ\_pliku*]  
 LOAD *spec\_pliku*  
 MERGE *spec\_pliku*  
 CHAIN *spec\_pliku* [,*wyr\_nr\_linii*]  
 CHAIN MERGE *spec\_pliku* [, *wyr\_nr\_linii*]  
 [DELETE *zakres\_linii*]

*Definicje i deklaracje*

DEFREAL *zakres\_liter*  
 DEFINT *zakres\_liter*  
 DEFSTR *zakres\_liter*  
 DIM *nazwa\_tabl* (*rozm* [,*rozm*]...) [,*nazwa\_tabl* (*rozmiary*)...]  
 ERASE *nazwa\_tabl* [,*nazwa\_tabl*]...  
 DEF FN*nazwa\_fun* [(*par* [,*par*]...) = wyrażenie

*Obliczenia arytmetyczne*

FN*nazwa\_fun\_aryt* [(*arg* [,*arg*]...)]  
 ABS (*wyr\_aryt*)  
 ATN (*wyr\_aryt*)  
 COS (*wyr\_aryt*)  
 EXP (*wyr\_aryt*)  
 FIX (*wyr\_aryt*)

INT (*wyr\_\_aryt*)  
 LOG (*wyr\_\_aryt*)  
 LOG10 (*wyr\_\_aryt*)  
 MAX (*wyr\_\_aryt* [,*wyr\_\_aryt*]...)  
 MIN (*wyr\_\_aryt* [,*wyr\_\_aryt*]...)  
 PI  
 ROUND (*wyr\_\_aryt* [,*wyr\_\_calk*])  
 SGN (*wyr\_\_aryt*)  
 SIN (*wyr\_\_aryt*)  
 SQR (*wyr\_\_aryt*)  
 TAN (*wyr\_\_aryt*)  
 CINT (*wyr\_\_aryt*)  
 CREAL (*wyr\_\_aryt*)  
 VAL (*wyr\_\_tekst*)  
 RND [(*wyr\_\_aryt*)]  
 TIME  
 [LET] *zm\_\_aryt* = *wyr\_\_aryt*  
 RANDOMIZE [*wyr\_\_aryt*]  
 RANDOMIZE TIME  
 RAD  
 DEG

### Przetwarzanie tekstów

FN*nazwa\_\_fun\_\_tekst* [(*arg* [,*arg*...])]  
 ASC (*wyr\_\_tekst*)  
 BIN\$ (*wyr\_\_calk*, *dlg\_\_tekstu*)  
 CHR\$ (*kod\_\_znaku*)  
 DEC\$ (*wyr\_\_aryt*, *wzorzec*)  
 HEX\$ (*wyr\_\_calk*, *dlg\_\_tekstu*)  
 INSTR ((*nr\_\_pozycji*,) *wyr\_\_tekst*, *wyr\_\_tekst*)  
 LEFT\$ (*wyr\_\_tekst*, *dlg\_\_tekstu*)  
 LEN (*wyr\_\_tekst*)  
 LOWER\$ (*wyr\_\_tekst*)  
 MID\$ (*wyr\_\_tekst*, *nr\_\_pozycji* [,*dlg\_\_tekstu*])  
 RIGHT\$ (*wyr\_\_tekst*, *dlg\_\_tekstu*)  
 SPACE\$ (*dlg\_\_tekstu*)  
 STR\$ (*wyr\_\_aryt*)  
 STRING\$ (*dlg\_\_tekstu*, *kod\_\_znaku*)  
 STRING\$ (*dlg\_\_tekstu*, *wyr\_\_tekst*)  
 UPPER\$ (*wyr\_\_tekst*)  
 [LET] *zm\_\_tekst* = *wyr\_\_tekst*  
 MID\$ (*zm\_\_tekst*, *nr\_\_pozycji* [,*dlg\_\_tekstu*]) = *wyr\_\_tekst*

## Sterowanie programem

```

■ REMAIN (nr__stopera)
END
STOP
GOTO nr__linii
IF wyr__log GOTO nr__linii [ELSE nr__linii_lub__instrukcje]
IF wyr__log THEN nr__linii_lub__instrukcje
                                [ELSE nr__linii_lub__instrukcje]
FOR zm__ster = wart__pocz TO wart__końc [STEP wart__kroku]
NEXT [zm__ster [,zm__ster]...]
WHILE wyr__log
WEND
GOSUB nr__linii
RETURN
ON wyr__calk GOTO nr__linii [,nr__linii]...
ON wyr__calk GOSUB nr__linii [,nr__linii]...
AFTER opóźnienie [,nr__stopera] GOSUB nr__linii
EVERY okres [,nr__stopera] GOSUB nr__linii
DI
EI
ON BREAK STOP
ON BREAK GOSUB nr__linii

```

## Wprowadzanie danych

```

■ INKEY (nr__klawisza)
INKEY$
INPUT [#nr__str,] [;] [”tekst__zachęty” sep] zmienna [,zmienna]...
LINE INPUT [#nr__str,] [;] [”tekst__zachęty” sep] zm__tekst
DATA stała [,stała]...
READ zmienna [,zmienna]...
RESTORE [nr__linii]
KEY nr__klawisza, wyr__tekst
KEY DEF nr__klawisza, par01 [,kod__ster]...
SPEED KEY opóźnienie, okres

```

## Wyprowadzanie wyników

```

■ POS (#nr__str)
VPOS (#nr__str)
MODE par012
BORDER nr__koloru__ramki [,nr__koloru__ramki]
INK nr__atramentu, nr__koloru [,nr__koloru]
SPEED INK okres, okres

```

PAPER [#nr\_\_str,] nr\_\_atramentu  
 PEN [#nr\_\_str,] nr\_\_atramentu  
 WINDOW [#nr\_\_str,] lewy, prawy, góra, dół  
 WINDOW SWAP nr\_\_str, nr\_\_str  
 CLS [#nr\_\_str]  
 LOCATE [#nr\_\_str,] nr\_\_wiersza, nr\_\_kol  
 PRINT [#nr\_\_str,] [wyrażenie [sep wyrażenie]...]  
     [USING wzorzec; wyrażenie [sep wyrażenie]...] [sep]  
     ■ SPC (liczba\_\_spacji)  
     ■ TAB (nr\_\_kol)  
 WIDTH szer\_\_wiersza  
 ZONE szer\_\_strefy  
 WRITE [#nr\_\_str,] wyrażenie [,wyrażenie]...  
 SYMBOL kod\_\_znaku, opis\_\_znaku  
 SYMBOL AFTER kod\_\_znaku

### Grafika

■ XPOS  
 YPOS  
 TEST (wsp\_\_x, wsp\_\_y)  
 ■ TESTR (przes\_\_x, przes\_\_y)  
 ORIGIN wsp\_\_x, wsp\_\_y [,lewy, prawy, góra, dół]  
 CLG [nr\_\_atramentu]  
 MOVE wsp\_\_x, wsp\_\_y  
 MOVER przes\_\_x, przes\_\_y  
 PLOT wsp\_\_x, wsp\_\_y [,nr\_\_atramentu]  
 PLOTTR przes\_\_x, przes\_\_y [,nr\_\_atramentu]  
 DRAW wsp\_\_x, wsp\_\_y [,nr\_\_atramentu]  
 DRAWR przes\_\_x, przes\_\_y [,nr\_\_atramentu]  
 TAG [#nr\_\_str]  
 TAGOFF [#nr\_\_str]

### Współpraca z plikami

■ EOF  
 OPENIN spec\_\_pliku  
 OPENOUT spec\_\_pliku  
 CAT  
 SPEED WRITE par01  
 CLOSEIN  
 CLOSEOUT

*Obsługa sytuacji błędnych*

■ ERL  
 ERR  
 ERROR *nr\_błędu*  
 ON ERROR GOTO *nr\_linii*  
 ON ERROR GOTO 0  
 RESUME [*nr\_linii*]  
 RESUME NEXT

*Łącze z zapisem wewnętrznym programu*

■ FRE (0)  
 FRE (" ")  
 HIMEM  
 INP (*nr\_portu*)  
 PEEK (*wyr\_adr*)  
 UNT (*wyr\_adr*)  
 VARPTR (*zmienna*)  
 VARPTR (#*nr\_str*)  
 CLEAR  
 CALL *wyr\_adr* [,*wyr\_aryt\_lub\_*(*a zmienna*)...  
 MEMORY *wyr\_adr*  
 POKE *wyr\_adr*, *wyr\_calk*  
 OUT *nr\_portu*, *wyr\_calk*  
 WAIT *nr\_portu* , *wyr\_calk* [,*wyr\_calk*]  
 SAVE *spec\_pliku* , *B*, *wyr\_adr*, *wyr\_adr* [,*wyr\_adr*]  
 LOAD *spec\_pliku*, *wyr\_adr*

*Generowanie dźwięku*

■ SQ (*par124*)  
 ENT *parametry\_opisu\_melodii*  
 ENV *parametry\_opisu\_melodii*  
 SOUND *parametry\_opisu\_melodii*  
 RELEASE *par1..7*  
 ON SQ (*par124*) GOSUB *nr\_linii*

*Współpraca z innymi urządzeniami*

■ JOY (*par01*)

### Wykaz komunikatów błędów

- 10 Array already dimensioned  
Próba powtórnej deklaracji tej samej tablicy.
- 17 Cannot CONTInue  
Niemożliwe wznowienie działania programu zleceniem CONTInue, gdyż dokonano modyfikacji treści programu.
- 4 DATA exhausted  
Brak danych dla instrukcji READ.
- 21 Direct command found  
Podczas ładowania programu z kasety wykryto linię bez numeru linii traktowaną jako zlecenie.
- 11 Division by zero  
Podczas wykonywania jednej z operacji:/\MOD lub↑ wynik przekracza zakres wartości liczbowej.
- 24 EOF met  
Próba odczytu z pamięci kasetowej następnej informacji po wykryciu znacznika końca pliku.
- 27 File already open  
Próba otwarcia jeszcze jednego pliku wejściowego lub wyjściowego bez zamknięcia dotychczas otwartego.
- 25 File type error  
Niezgodność typu pliku z próbą jego wykorzystania.
- 5 Improper argument  
Ogólny komunikat błędu dotyczący niezgodności argumentów wywoływanej funkcji z jej parametrami lub błędów w parametrach polecenia.
- 12 Invalid direct command  
Wprowadzone polecenie nie może być wykonane natychmiast jako zlecenie.
- 8 Line does not exist  
Brak linii o danym numerze.
- 23 Line too long  
Wprowadzona linia programu po przetworzeniu w systemie Basic jest za długa.
- 7 Memory full  
Tekst programu i obszar pamięci potrzebny na jego zmienne oraz wykonanie nie mieści się w dostępnej pamięci operacyjnej.



- 26 **NEXT missing**  
W programie brak instrukcji NEXT domykającej instrukcję cyklu wyliczanego rozpoczynającego się instrukcją FOR.
- 22 **Operand missing**  
Nieprawidłowa składnia wyrażenia.
- 6 **Overflow**  
Wynik operacji arytmetycznych przekracza dopuszczalny zakres wartości liczb rzeczywistych lub podczas konwersji zakres wartości liczb całkowitych.
- 19 **RESUME missing**  
Brak instrukcji RESUME kończącej podprogram obsługi sytuacji błędnych.
- 16 **String expression too complex**  
Zbyt skomplikowane obliczenia w wyrażeniu tekstowym dające za dużo pośrednich wartości tekstowych.
- 14 **String space full**  
Brak miejsca na przechowywanie nowych wartości tekstowych.
- 15 **String too long**  
Długość tekstu przekracza 255 znaków.
- 9 **Subscript out of range**  
Jeden ze wskaźników zmiennej wskaźnikowej przekracza zadeklarowany zakres rozmiaru wymiaru tablicy.
- 2 **Syntax error**  
Nierozumiały zapis polecenia — błędy w składni.
- 13 **Type mismatch**  
Niezgodność typów wartości między argumentami a odpowiadającymi im parametrami lub błąd w zapisie wartości dla instrukcji wprowadzania READ lub INPUT.
- 1 **Unexpected NEXT**  
Próba wykonania instrukcji NEXT, dla której nie ma odpowiadającej jej instrukcji początku cyklu wyliczanego FOR.
- 20 **Unexpected RESUME**  
Próba wykonania instrukcji RESUME poza podprogramem obsługi sytuacji błędnej.
- 3 **Unexpected RETURN**  
Próba wykonania instrukcji RETURN, dla której nie było instrukcji wywołania podprogramu GOSUB.

- 30 Unexpected WEND  
Próba wykonania instrukcji WEND, dla której nie ma odpowiadającej jej instrukcji początku cyklu z warunkiem WHILE.
- 28 Unknown command  
Nieznane polecenie
- 18 Unknown user function  
Brak definicji funkcji dla danego wywołania.
- 29 WEND missing  
W programie brak instrukcji WEND domykającej cykl z warunkiem wstępnym — rozpoczęty instrukcją WHILE.

## 3. Apple II

Mikrokomputer Apple IIe jest trzecim modelem komputerów tej serii z nieco zmodyfikowaną klawiaturą i dodatkowym wyposażeniem.

W mikrokomputerach tych są dostępne dwie wersje języka Basic — Integer BASIC oraz Applesoft BASIC. Sposób wyboru jednej z wersji jest zależny od typu mikrokomputera lub od wywołania systemu BASIC. Integralnym elementem funkcjonalnym Apple'a jest system operacyjny DOS ładowany automatycznie z 5,25-calowej dyskietki po włączeniu komputera (w niektórych wersjach trzeba wymusić załadowanie systemu). Interpretatory języka BASIC mogą być umieszczone w pamięci stałej ROM lub muszą być ładowane z dyskietki.

Niekiedy jeden z interpretatorów znajduje się w pamięci stałej, a drugi musi być ładowany. Rozmieszczenie interpretatorów jest uzależnione od wersji mikrokomputera i zainstalowanych w nim dodatkowych kart.

### *Aadres*

Jeżeli występuje to określa *adres* różny od zera w przestrzeni adresowej pamięci operacyjnej.

### *Bn*

Jeżeli występuje, to określa numer bajtu różny od zera w pliku sekwencyjnym.

### *C, I, O*

Jeżeli występują to włączają wyświetlanie na ekranie:

*C* — zleceń dla systemu DOS wywoływanych z programu,

*I* — danych wprowadzanych z pliku wejściowego,

*O* — wyników wyprowadzanych do pliku wyjściowego.

### *Dn, Sn, Vn*

Identyfikacja numeru dysku D1 lub D2, sprzęgu S0..S7 i numeru woluminu V0..V255.

*deklaracja tablicy (I)\*)*

Możliwa jest deklaracja tablicy liczbowej i tekstowej tylko jedno-wymiarowej o maksymalnej wartości wskaźnika 255, licząc od wartości minimalnej równej 0. Zmienna wskaźnikowa o wskaźniku równym 0 jest tożsama ze zmienną prostą o tym samym identyfikatorze. Tablica typu tekstowego przechowuje w każdym elemencie pojedynczy znak tekstu o maksymalnej długości tekstu równej zadeklarowanemu rozmiarowi tablicy + 1.

*deklaracja tablicy (A)*

Tablice mogą mieć do 88 wymiarów. Dopuszczalna jest dowolna wartość dodatnia wskaźnika. Istnieje również deklaracja standardowa tablic niezadeklarowanych.

*dlg\_\_rek*

Stała całkowita z zakresu 1..32767 określająca w liczbie bajtów długość rekordu dla plików o dostępie swobodnym.

*identyfikacja pliku*

Tekst zawierający do 30 znaków — z wyjątkiem znaku przecinka — rozpoczynający się od litery.

*identyfikator (I)*

Identyfikatorem jest ciąg do 100 znaków — złożony z liter i cyfr — rozpoczynający się od litery. Identyfikatory słów kluczowych są zastrzeżone.

*identyfikator (A)*

Identyfikatorem jest ciąg do 40 znaków — złożony z liter i cyfr, rozpoczynający się od litery — przy czym **rozdzielalne są tylko 2 pierwsze znaki**. Identyczne identyfikatory mogą być wykorzystane dla różnych → *typów zmiennych i funkcji*. Identyfikatory słów kluczowych są zastrzeżone.

*ekran graficzny (I + A)*

Monitor ekranowy jest przełączany w tryb graficzny o niskiej rozdzielczości poleceniem:

GR — 40 × 48 punktów w 16 kolorach z → *palety barw*.

Ostatnie 4 wiersze → *ekranu tekstowego* mogą być przełączone w tryb graficzny (dla *wsp\_\_y*: 40..47) poleceniem POKE — 16302,0 i następnie

---

\*) Dla rozróżnienia opisów elementów jednej lub obu wersji języka zastosowano oznaczenia: (I) — Integer BASIC, (A) — Applesoft BASIC, (I + A) — obie wersje.

odtworzone jako okno tekstowe poleceniem POKE – 16301,0. Zakres zmian współrzędnych ekranu jest określony w poziomie *wsp\_\_x*: 0..39 oraz w pionie *wsp\_\_y*: 0..47 lub 0..39, ze standardowym położeniem punktu początkowego układu w lewym górnym rogu ekranu.

#### *ekran graficzny (A)*

Monitor ekranowy może być dodatkowo przełączony w tryb graficzny o wysokiej rozdzielczości poleceniem:

HGR — 280 × 192 punkty w jednym z 8 kolorów według dwóch HGR2 rozłącznych czterokolorowych → *palet barw* (HGR2 udostępnia na ekranie drugą stronę).

Ostatnie 4 wiersze (dla *wsp\_\_y*: 159..191) mogą być przełączone w tryb graficzny jak dla niskiej rozdzielczości. Zakres zmian współrzędnych ekranu jest określony w poziomie *wsp\_\_x*: 0..279 oraz w pionie *wsp\_\_y*: 0..159 lub 0..191 z punktem początkowym w lewym górnym rogu ekranu. Do wypełniania ekranu graficznego o wysokiej rozdzielczości, oprócz instrukcji graficznych, wykorzystuje się definicję kształtu obrazów lokalizowanych bezpośrednio w pamięci. Obrazy te mogą być następnie szybko wyświetlane z możliwością przeskalowania ich wielkości oraz rotacji.

#### *ekran tekstowy (I + A)*

Monitor ekranowy znajduje się standardowo w trybie tekstowym lub może być do niego przełączony z trybu ekranu graficznego zleceniem:

TEXT — 24 wiersze po 40 lub 80 znaków zależnie od wyposażenia sprzętowego i wydania zlecenia PR#3.

Zakres numeracji wierszy jest określony w przedziale 1..24, a kolumn w przedziale 1..40 lub 1..80. Zmianę rozmiaru okna logicznego ekranu tekstowego można uzyskać wymieniając wartości bajtów w pamięci:

POKE 32, *lewy\_\_margines* w zakresie 0..39 lub 0..78

POKE 33, *szer\_\_wiersza* w zakresie 1..40 lub 2..80

POKE 34, *górn\_\_wiersz* w zakresie 0..23

POKE 35, *liczba\_\_wierszy* w zakresie 0..23

POKE 36, *poz\_\_kursora\_\_w\_\_wierszu*

POKE 37, *poz\_\_kursora\_\_w\_\_kol*

**Uwaga:** Należy uważać, aby nowe parametry nie spowodowały definicji okna tekstowego poza ekranem rzeczywistym.

#### *klawiatura*

Klawiatura jest ułożona w standardowym układzie QWERTY klawiszy wprowadzających kody znaków dużych i małych (małe tylko Apple IIe) liter, cyfr oraz symboli i znaków sterujących.

*klawisze sterujące*

Wyróżnione są następujące klawisze o kodach znaków sterujących:

- RETURN — zakończenie wprowadzania lub poprawiania linii zawierającej polecenie.
- SHIFT — wprowadzenie dużej litery (tylko Apple IIe) lub symbolu wskazanego równocześnie wciśniętym klawiszem.
- CAPS\_LOCK — przełączenie na stałe wprowadzania liter tylko jako dużych (istnieje tylko w wersji Apple IIe) aż do ponownego wciśnięcia CAPS\_LOCK.
- CONTROL — wprowadzenie znaku sterującego związanego z równocześnie wciśniętym klawiszem.
- ESC — przełączenie na tryb wprowadzania awaryjnego sekwencji kolejnych znaków do czasu ponownego wciśnięcia ESC.
- RESET — przerwanie aktualnie wykonywanego zlecenia lub programu — czasem konieczne jest wprowadzenie CONTROL&RESET.
- — przesuwanie w prawo kursora z kopiowaniem do bufora znaku z ekranu.
- ← — przesuwanie wstecz kursora z wymazaniem znaku w buforze bez usunięcia go z ekranu.

Pozostałe funkcje sterujące uzyskuje się wykorzystując kombinacje klawiszy sterujących z innymi klawiszami:

**CONTROL&RESET**

- przerwanie wykonywania zlecenia lub programu i przełączenie na wprowadzanie nowej linii polecenia.

**CONTROL&OPEN\_APPLE&RESET**

- wyzerowanie stanu komputera (OPEN\_APPLE — jasne jabłuszko — tylko Apple IIe).

- przesuwanie kursora po ekranie w trybie awaryjnym:\*)

ESC A — w prawo,

ESC C — na dół,

ESC B — w lewo,

ESC D — w górę.

- przesuwanie kursora po ekranie w trybie awaryjnym po ESC aż do następnego ESC (tylko Apple IIe):

→ — w prawo,

↓ — na dół,

← — w lewo,

↑ — w górę.

\*) Tylko dla wersji z 40 znakami w wierszu.

- przesuwanie kursora po ekranie w trybie awaryjnym po ESC aż do następnego ESC (dla Apple IIe patrz\*)
- K — w prawo, M — na dół
- J — w lewo, I — w górę.
- CONTROL&X — usunięcie treści aktualnie wprowadzonej linii.
- ESC E — skasowanie części linii od pozycji kursora do jej końca\*).
- ESC F — skasowanie części okna tekstowego od pozycji kursora do ostatniego znaku w oknie\*).
- ESC @ — czyszczenie okna tekstowego.

*linia programu (I)*

W linii może być do 150 znaków. Zakres numeracji linii wynosi 1..32 767.  
W linii będącej zleceniem dopuszczalna jest tylko pojedyncza instrukcja.

*linia programu (A)*

W linii może być do 255 znaków. Zakres numeracji linii wynosi 0..63 999.

*nr\_obrazu (A)*

Wyrażenie całkowite o wartości określającej numer obrazu zdefiniowanego w pamięci. Zakres numeracji obrazów jest zależny od liczby ich definicji — nie może przekraczać 255.

*operatory (I + A)*

- arytmetyczne:  $\uparrow$  - \* / MOD (tylko dla I) + -
- relacji (I) : = < <= # <> >= >
- relacji (A) : = < <= = < <> > < == > >= >
- logiczne : NOT AND OR
- tekstowe (I) : = # <>
- tekstowe (A): + oraz relacji

Operator # jest operatorem nierówności.

*opis zakresu numeracji linii (I + A)*

Zakres numeracji linii określa fragment programu, który ma być poddany działaniu zlecenia i może przyjmować postać:

- nr\_linii* — tylko linia o podanym numerze,
- nr\_linii, nr\_linii* — fragment od pierwszej podanej linii do drugiej.

---

\*) Tylko dla wersji z 40 znakami w wierszu.

*opis zakresu numeracji linii (A)*

Dodatkowo dla zlecenia LIST mogą być określone zakresy:

pusty	— cały tekst programu,
<i>nr_linii</i> -	— fragment od danej linii do końca tekstu programu,
- <i>nr_linii</i>	— fragment od pierwszej linii do podanej,
<i>nr_linii</i> - <i>nr_linii</i>	— fragment od pierwszej podanej linii do drugiej.

*paleta barw (I + A)*

Dla monitora kolorowego pracującego w trybie graficznym o niskiej rozdzielczości jest dostępnych 16 kolorów wybieranych wartością wyrażenia całkowitego z zakresu 0..255 modulo 16:

0 czarny	6 średnioniebieski	11 różowy
1 karminowy	7 jasnoniebieski	12 jasnozielony
2 ciemnoniebieski	8 brązowy	13 żółty
3 purpurowy	9 pomarańczowy	14 wodny
4 ciemnozielony	10 szary#2	15 biały
5 szary#1		

*paleta barw (A)*

Dla monitora kolorowego pracującego w trybie graficznym o wysokiej rozdzielczości są dodatkowo dostępne, w każdej 7-punktowej strefie pionowej, kolory z jednej spośród dwóch palet:

0 czarny	4 czarny
1 zielony	5 pomarańczowy
2 purpurowy	6 niebieski
3 biały	7 biały

*poprawianie linii programu (I + A)*

Pojedyncze linie programu w trakcie wprowadzania lub po wylistowaniu ich treści na ekranie poleceniem LIST mogą być poprawiane następująco:

- przesunąć kursor na ekranie na początek poprawianej linii wykorzystując w trybie awaryjnym po ESC klawisze ↑ ← → ↓ (lub I, J, K, M) aż do wyłączenia trybu klawiszem ESC po osiągnięciu żądanej pozycji,
- kopiować poza trybem awaryjnym poprawne fragmenty tekstu linii klawiszem →,
- omijać niepoprawne fragmenty tekstu w trybie awaryjnym po ESC wykorzystując klawisze ← → aż do wyłączenia trybu klawiszem ESC,
- wymieniać niepoprawne fragmenty tekstu, usuwając zbędne znaki klawiszem ← i wstawiając na te miejsca nowe znaki,



— wstawiać nowe fragmenty w tekst linii stosując — począwszy od znaku bezpośrednio mającego nastąpić po wstawianym tekście — sekwencje:

- przesunięcia kursora w trybie awaryjnym do linii poprzedniej w górę ESC ↑ ESC,
- skasowanie części linii do jej końca ESC E,
- wpisanie nowego tekstu np. ALFA,
- powrót do trybu awaryjnego do znaku bezpośrednio następnego po nowym fragmencie ESC ↓ ← ← ← ← ESC;

— skopiować pozostałe znaki do końca linii,

— zakończyć poprawianie linii klawiszem RETURN.

W trybie awaryjnym zamiast sekwencji ESC z pojedynczym znakiem można lepiej używać sekwencji ESC A, ESC B, itp.

### Rn

Jeżeli występuje, to określa — licząc od zera — numer linii *n* w pliku, poczynając od którego mają być pobierane polecenia.

### rodzaj pliku (I + A)

W pamięci dyskowej możliwe jest utworzenie plików o organizacji sekwencyjnej i o dostępie bezpośrednim.

### słowa kluczowe (I + A)

Identyfikatory słów kluczowych są zastrzeżone i są w pierwszej kolejności wyszukiwane w treści polecenia.

### szer\_\_strefy (I)

Szerokość strefy dla polecenia wyprowadzania PRINT wynosi 8 znaków.

### szer\_\_strefy (A)

Szerokość strefy dla polecenia wyprowadzania PRINT wynosi 16 znaków.

### typ zmiennej (I)

Istnieją dwa typy zmiennych rozróżnianych istnieniem znacznika typu:

- liczbowy całkowity (bez znacznika),
- \$ — tekstowy.

### typ zmiennej i funkcji (A)

Istnieją zmienne trzech typów rozróżnianych znacznikiem typu:

- % — liczbowy całkowity,
- liczbowy rzeczywisty (bez znacznika),
- \$ — tekstowy.

*wartość liczbowa (I)*

Dopuszczalne są jedynie wartości liczbowe całkowite z zakresu  $-32\,767..32\,767$ .

*wartość liczbowa (A)*

Dopuszczalne są wartości liczbowe typu:

- całkowitego z zakresu  $-32\,767..32\,767$  w postaci liczby całkowitej,
- rzeczywistego z zakresu

$$-1,7 \cdot 10^{38} .. -3 \cdot 10^{-38}, 0, 3 \cdot 10^{-38} .. 1,7 \cdot 10^{38}$$

z dokładnością do 9 cyfr znaczących w postaci liczby całkowitej, ułamkowej i wykładniczej.

*wartość logiczna (I + A)*

Wartość logiczna — prawda (1), fałsz (0).

*wartość tekstowa (I)*

Dopuszczalna jest jedynie wartość tekstowa zapisana w tablicy po jednym znaku w każdym elemencie.

*wartość tekstowa (A)*

Teksty mogą zawierać od 0 do 255 znaków o wartości kodu z zakresu 0..127, z wyjątkiem kodów znaków  $\uparrow \leftarrow \rightarrow \downarrow$  CONTROL&H CONTROL&M CONTROL&U CONTROL&X ESC RETURN.

*wywołanie systemu Basic*

Po włączeniu mikrokomputera (lub w niektórych modelach po załadowaniu systemu) zgłasza się bezpośrednio system Basic z jedną z wersji języka rozróżnianych systemowym znakiem zachęty:

> Integer BASIC

] Applesoft BASIC

Przełączenie na właściwą wersję języka następuje po uruchomieniu programu lub też należy użyć zlecenia INT dla wywołania wersji Integer oraz FP dla wersji Applesoft. Istnieją też różne możliwości rozmieszczenia w pamięci ROM lub w pamięci dyskowej jednej lub obu wersji języka.

*zaokrąglenia*

Wartość liczbowa rzeczywista w przypadku potrzeby wykorzystania jej jako całkowitej jest zaokrąglana przez obcięcie części ułamkowej z otrzymaniem wartości całkowitej nie większej od danej wartości rzeczywistej (entier).

### Wykaz poleceń systemu DOS

FP [,Dn] [,Sn] [,Vn]

INIT *spec\_pliku* [,Dn] [,Sn] [,Vn]

wykorzystanie zlecenia DOS w programie:

*nr\_linii* PRINT "<CTRL-D>"; "*treść\_zlecenia\_DOS*"

*treść\_zlecenia\_DOS*

BLOAD *spec\_pliku* [,Aadres] [,Dn] [,Sn] [,Vn]

BRUN *spec\_pliku* [,Aadres] [,Dn] [,Sn] [,Vn]

BSAVE *spec\_pliku* [,Aadres] [,Ldlg\_rek] [,Dn] [,Sn] [,Vn]

CATALOG [,Dn] [,Sn]

CHAIN *spec\_pliku* [,Dn] [,Sn] [,Vn]

CLOSE *spec\_pliku*

DELETE *spec\_pliku* [,Dn] [,Sn] [,Vn]

EXEC *spec\_pliku* [,Rn] [,Dn] [,Sn] [,Vn]

IN# *nr\_portu*

LOAD *spec\_pliku* [,Dn] [,Sn] [,Vn]

LOCK *spec\_pliku* [,Dn] [,Sn] [,Vn]

MAXFILES *stała\_calk*

MON [C] [,I] [,O]

NOMON [C] [,I] [,O]

PR# *nr\_portu*

RUN *spec\_pliku* [,Dn] [,Sn] [,Vn]

SAVE *spec\_pliku* [,Dn] [,Sn] [,Vn]

UNLOCK *spec\_pliku* [,Dn] [,Sn] [,Vn]

VERIFY *spec\_pliku* [,Dn] [, Sn] [,Vn]

APPEND *spec\_pliku* [,Dn] [,Sn] [,Vn]

OPEN *spec\_pliku* [,dlg\_rek] [,Dn] [,Sn] [,Vn]

POSITION *spec\_pliku* [,Rn]

READ *spec\_pliku* [,Rn] [,Bn]

WRITE *spec\_pliku* [,Rn] [,Bn]

### Wykaz poleceń — Integer BASIC

*Edycja programu*

INT

NEW

AUTO *nr\_linii* [,inkrement]

<CTRL-X>MAN

'wstrzymanie AUTO

DEL *zakres\_linii*

LIST *zakres\_linii*

*Komentarze*

REM *dowolny\_tekst*

*Uruchomienie i śledzenie programu*

RUN [*nr\_linii*]  
 RUN *wyr\_calk*  
 CON  
 TRACE  
 NOTRACE  
 DSP *zmienna*  
 NODSP *zmienna*

*Ladowanie, skladowanie i wiązanie programów*

SAVE  
 LOAD  
 (patrz wykaz poleceń systemu DOS)

*Definicje i deklaracje*

DIM *nazwa\_tabl (rozmiar) [,nazwa\_tabl(rozmiar)]...*

*Obliczenia arytmetyczne*

ABS (*wyr\_calk*)  
 SGN (*wyr\_calk*)  
 RND (*wyr\_calk*)  
 [LET] *zm\_aryt* = *wyr\_calk*

*Przetwarzanie tekstów*

ASC (*wyr\_tekst*)  
 LEN (*wyr\_tekst*)  
 [LET] *zm\_tekst* = *wyr\_tekst*

*Sterowanie programem*

GOTO *nr\_linii*  
 GOTO *wyr\_nr\_linii*  
 IF *wyr\_log* THEN [GOTO] *nr\_linii*  
 IF *wyr\_log* THEN *instrukcja*  
 POP  
 END  
 FOR *zm\_ster* = *wart\_pocz* TO *wart\_konc* [STEP *wart\_kroku*]  
 NEXT *zm\_ster* [,*zm\_ster*]...  
 GOSUB *nr\_linii*  
 GOSUB *wyr\_nr\_linii*  
 RETURN

*Wprowadzenie danych*

INPUT ["*tekst\_zachety*",] *zmienna* [,*zmienna*]...

*Wyprowadzanie wyników*

TEXT

PRINT [*wyrażenie* [*sep* *wyrażenie*]...] [*sep*]  
 TAB *nr\_\_kol*  
 VTAB *nr\_\_wiersza*

### Grafika

■ SCRN (*wsp\_\_y,wsp\_\_x*)  
 GR  
 COLOR = *nr\_\_koloru*  
 PLOT *wsp\_\_y,wsp\_\_x*  
 HLIN *wsp\_\_y,wsp\_\_y* AT *wsp\_\_x*  
 VLIN *wsp\_\_x,wsp\_\_x* AT *wsp\_\_y*

### Łącze z zapisem wewnętrznym programu

■ PEEK (*wyr\_\_adr*)  
 CALL *wyr\_\_adr*  
 POKE *wyr\_\_adr,wyr\_\_calk*  
 CLR  
 HIMEM: *wyr\_\_adr*  
 LOMEM: *wyr\_\_adr*

### Współpraca z innymi urządzeniami

■ PDL (*par0123*)  
 IN# *nr\_\_portu*  
 PR# *nr\_\_portu*

## Wykaz poleceń — Applesoft BASIC

### Edycja programu

NEW  
 DEL *zakres\_\_linii*  
 LIST *zakres linii*

### Komentarze

REM *dowolny\_\_tekst*

### Uruchomienie i śledzenie programu

RUN [*nr\_\_linii*]  
 CONT  
 TRACE  
 NOTRACE

### Ładowanie, składowanie i wiązanie programów (patrz wykaz poleceń systemu DOS)

*Definicje i deklaracje*

DIM nazwa\_\_tabl (rozmi[rozmi]...) [,nazwa\_\_tabl(rozmiary)]...  
 DEF FNnazwa\_\_fun (par) = wyr\_\_aryt

*Obliczenia arytmetyczne*

FN nazwa\_\_fun (wyr\_\_aryt)  
 ABS (wyr\_\_aryt)  
 ATN (wyr\_\_aryt)  
 COS (wyr\_\_aryt)  
 EXP (wyr\_\_aryt)  
 INT (wyr\_\_aryt)  
 LOG (wyr\_\_aryt)  
 SGN (wyr\_\_aryt)  
 SIN (wyr\_\_aryt)  
 SQR (wyr\_\_aryt)  
 TAN (wyr\_\_aryt)  
 VAL (wyr\_\_tekst)  
 RND (wyr\_\_aryt)  
 [LET] zm\_\_aryt = wyr\_\_calk

*Przetwarzanie tekstów*

ASC (wyr\_\_tekst)  
 CHR\$ (kod\_\_znaku)  
 LEFT\$ (wyr\_\_tekst,dlg\_\_tekstu)  
 LEN (wyr\_\_tekst)  
 MID\$ (wyr\_\_tekst,nr\_\_pozycji [,dlg\_\_tekstu])  
 RIGHT\$ (wyr\_\_tekst,dlg\_\_tekstu)  
 STR\$ (wyr\_\_aryt)  
 [LET] zm\_\_tekst = wyr\_\_tekst

*Sterowanie programem*

END  
 STOP  
 GOTO nr\_\_linii  
 IF wyr\_\_log THEN [GOTO] nr\_\_linii  
 IF wyr\_\_log THEN instrukcje  
 FOR zm\_\_ster = wart\_\_pocz TO wart\_\_końc [STEP wart\_\_kroku]  
 NEXT [zm\_\_ster [,zm\_\_ster]...]  
 GOSUB nr\_\_linii  
 POP  
 RETURN  
 ON wyr\_\_calk GOTO nr\_\_linii [,nr\_\_linii]...  
 ON wyr\_\_calk GOSUB nr\_\_linii [,nr\_\_linii]...

*Wprowadzanie danych*

GET *zmienna*  
 INPUT [”*tekst\_zachęty*”,] *zmienna* [,*zmienna*]...  
 DATA *stała* [,*stała*]...  
 READ *zmienna* [,*zmienna*]...  
 RESTORE

*Wyrowadzanie wyników*

■ POS(0)  
 TEXT  
 FLASH  
 INVERSE  
 NORMAL  
 HTAB *nr\_kol*  
 VTAB *nr\_wiersza*  
 PRINT [*wyrażenie* [*sep* *wyrażenie*]...] [*sep*]  
 ?[*wyrażenie* [*sep* *wyrażenie*]...] [*sep*]  
 ■ SPC (*liczba\_spacji*)  
 ■ TAB (*nr\_kol*)  
 SPEED *wyr\_calk*

*Grafika*

■ SCRN (*wsp\_y,wsp\_x*)  
 GR  
 HGR  
 HGR2  
 HOME  
 COLOR = *nr\_koloru*  
 HCOLOR = *nr\_koloru*  
 SCALE = *wyr\_calk*  
 ROT = *wyr\_calk*  
 PLOT *wsp\_y,wsp\_x*  
 HPLOT [TO] *wsp\_y,wsp\_x*  
 HPLOT *wsp\_y,wsp\_x* TO *wsp\_y,wsp\_x* [TO *wsp\_y,wsp\_x*]...  
 HLIN *wsp\_y,wsp\_y* AT *wsp\_x*  
 VLIN *wsp\_x,wsp\_x* AT *wsp\_y*  
 DRAW *nr\_obrazu* [AT *wsp\_y,wsp\_x*]  
 XDRAW *nr\_obrazu* [AT *wsp\_y,wsp\_x*]

*Współpraca z plikami na kasecie*

STORE *zm\_aryt*  
 RECALL *zm\_aryt*  
 SHLOAD

*Obsługa sytuacji błędnych*

ONERR GOTO *nr\_linii*  
RESUME

*Łączy z zapisem wewnętrznym programu*

■ PEEK (*wyr\_adr*)  
FRE(0)  
■ USR (*wyr\_aryt*)  
CLEAR  
CALL *wyr\_adr*  
POKE *wyr\_adr,wyr\_calk*  
WAIT *wyr\_adr,wyr\_calk* [*,wyr\_calk*]

*Współpraca z innymi urządzeniami*

■ PDL (*par0123*)

**Wykaz komunikatów błędów — System DOS**

- 9 DISK FULL  
Próba zapisania informacji na nośniku dyskowym przy braku na nim wolnego miejsca.
- 5 END OF DATA  
Próba odczytu informacji z pliku, gdy nie była ona wcześniej zapisana.
- 10 FILE LOCKED  
Próba dostępu do pliku z protekcją.
- 6 FILE NOT FOUND  
Na nośniku nie znaleziono pliku o podanym identyfikatorze,
- 13 FILE TYPE MISMATCH  
Nie zgodność typu pliku z próbą jego wykorzystania.
- 8 I/O ERROR  
Błąd w funkcjonowaniu urządzenia adaptera dyskietek podczas wykonywania operacji wejścia-wyjścia.
- 1 LANGUAGE NOT AVAILABLE  
Próba zmiany wersji języka Basic, gdy brak jest tej nowej wersji w pamięci ROM lub na nośniku dyskietki.
- 12 NO BUFFERS AVAILABLE  
Brak wolnego obszaru na bufor do współpracy z plikami.



- 15 NOT DIRECT COMMAND  
Zlecenia APPEND, OPEN, POSITION, READ i WRITE mogą być używane jedynie jako instrukcje zawarte w instrukcji PRINT.
- 14 PROGRAM TOO LARGE  
Podczas ładowania treści programu z pliku zabrakło wolnego obszaru w pamięci operacyjnej.
- 2 RANGE ERROR  
3 Wartość parametru podanego w zleceniu DOS jest nieprawidłowa.
- 11 SYNTAX ERROR  
Błąd w składni zlecenia DOS.
- 7 VOLUME MISMATCH  
Parametr numeru woluminu V nie jest zgodny z numerem aktualnie zainstalowanej dyskietki.
- 4 WRITE PROTECTED  
Próba zapisu na dyskietkę z protekcją zapisu.

### Wykaz komunikatów błędów — Integer BASIC

- \*\*\* > 255 ERR  
Wartość parametru poza zakresem 0..255.
- \*\*\* > 32 767 ERR  
Wartość liczby poza zakresem - 32 767..32 767.
- \*\*\* 16 FORS ERR  
Próba rozpoczęcia 17-ego równocześnie aktywnego cyklu wyliczanego.
- \*\*\* 16 GOSUBS ERR  
Próba wywołania kolejnego 17-ego podprogramu przy nieskończonych poprzednio wywołanych.
- \*\*\* BAD BRANCH ERR  
Próba przejścia do nieistniejącej linii programu.
- \*\*\* BAD NEXT ERR  
Próba wykonania instrukcji NEXT, dla której nie ma odpowiadającej jej instrukcji początku cyklu wyliczanego FOR.
- \*\*\* BAD RETURN ERR  
Próba wykonania instrukcji RETURN, dla której nie była wykonana instrukcja wywołania podprogramu GOSUB.
- \*\*\* DIM ERR  
Próba powtórnej deklaracji tej samej tablicy.

**\*\*\* MEM FULL ERR**

Za mało pamięci.

**\*\*\* NO END ERR**

Ostatnia wykonana instrukcja programu nie jest instrukcją END.

**\*\*\* RANGE ERR**

Wartość wskaźnika w zmiennej wskaźnikowej przekracza zadeklarowany zakres rozmiaru tablicy lub wartość argumentu w instrukcjach HLIN, VLIN, PLOT, TAB lub VTAB przekracza dopuszczalny zakres.

**RETYPE LINE**

Wprowadź powtórnie wartości dla instrukcji INPUT, gdyż poprzednie były zapisane błędnie.

**STOPPED AT *nr*\_\_*linii***

Komunikat informujący o miejscu zatrzymania działania programu wskutek wykrycia błędu, wykonania instrukcji END lub przerwania z klawiatury klawiszem CONTROL&C.

**\*\*\* STRING ERR**

Nielegalna operacja na tekstach.

**\*\*\* STR OVFL ERR**

Za długa wartość tekstowa w stosunku do zadeklarowanego rozmiaru tablicy tekstowej.

**\*\*\* SYNTAX ERR**

Błąd w składni zlecenia.

**\*\*\* TOO LONG ERR**

Za dużo par nawiasów (więcej niż 12) w wyrażeniu lub więcej niż 128 znaków w linii.

**Wykaz komunikatów błędów — Applesoft BASIC****107 ? BAD SUBSCRIPT ERROR**

Wartość jednego lub więcej wskaźników zmiennej wskaźnikowej przekracza zakres wartości rozmiaru danego wymiaru podanego w deklaracji tablicy.

**254 BAD RESPONSE TO AN INPUT**

Nieprawidłowy typ wartości danych wprowadzanych na żądanie instrukcji INPUT.

**BREAK IN *nr*\_\_*linii***

Komunikat podający miejsce działania programu wskutek wykonania STOP, END lub ostatniej instrukcji, lub też w wyniku przerwania z klawiatury klawiszem CONTROL&C.

- ? CANT CONTINUE ERROR  
Próba wznowienia działania programu poleceniem CONT podczas gdy program nie istnieje w pamięci, bądź też uprzednio został wykryty błąd fatalny albo też dokonano modyfikacji treści programu.
- 255 CONTROL-C HAS BEEN STRUCK  
Zatrzymanie programu w wyniku przerwania z klawiatury klawiszem CONTROL&C.
- 133 ? DIVISION BY ZERO ERROR  
W trakcie obliczeń wykryto próbę dzielenia przez wartość równą 0.
- 191 ? FORMULA TOO COMPLEX ERROR  
Zbyt skomplikowana składnia linii programu zawierająca instrukcję warunkową IF *wyr\_log* THEN...
- ? ILLEGAL DIRECT ERROR  
Zlecenia INPUT, DEF FN lub GET nie mogą być wykonywane natychmiast jako polecenia.
- 53 ? ILLEGAL QUANTITY ERROR  
Wartość liczbowa jest spoza zakresu w przypadku argumentu funkcji tekstowej lub arytmetycznej, oraz jako parametr instrukcji graficznych, itp.
- 0 ? NEXT WITHOUT FOR ERROR  
Brak odpowiedniej instrukcji początku cyklu wyliczanego dla aktualnie wykonywanej instrukcji NEXT.
- 42 ? OUT OF DATA ERROR  
Brak danych dla instrukcji READ.
- 77 ? OUT OF MEMORY ERROR  
Tekst programu i obszar pamięci potrzebny dla jego zmiennych oraz wykonania nie mieści się w dostępnej pamięci operacyjnej.
- 69 ? OVERFLOW ERROR  
Wynik operacji arytmetycznych przekracza dopuszczalny zakres wartości liczbowych rzeczywistych.
- 120 ? REDIM'D ARRAY ERROR  
Próba powtórnej deklaracji tablicy.
- 22 ? RETURN WITHOUT GOSUB ERROR  
Próba wykonania instrukcji RETURN, dla której nie było instrukcji wywołania podprogramu GOSUB.
- 176 ? STRING TOO LONG ERROR  
Długość tekstu przekracza 255 znaków.

- 16 ? SYNTAX ERROR  
Niezrozumiały zapis polecenia — błędy w składni.
- 163 ? TYPE MISMATCH ERROR  
Niezgodność typów wartości w instrukcji przypisania LET.
- 224 ? UNDEF'D FUNCTION ERROR  
Brak definicji funkcji dla danego wywołania.
- 90 ? UNDEF'D STATEMENT ERROR  
Brak linii o danym numerze.

## 4. IBM PC

Mikrokomputer osobisty IBM PC jest standardowo wyposażony w interpretator języka Basic. Jest to pełna wersja standardu przemysłowego języka BASIC Microsoft ze znacznymi rozszerzeniami w zakresie grafiki. Dla IBM PC są zaimplementowane trzy poziomy tego języka:

- **Cassette BASIC** wbudowany w pamięci ROM, działający bez adaptera dyskietek, współpracujący z pamięcią kasetową;
- **Disk BASIC** (ładowany z adaptera dyskietek 5,25 cala) działający pod kontrolą systemu operacyjnego MS-DOS lub CP/M i z nimi współpracujący;
- **Advanced BASIC** (BASICA-Version 2.0) będący rozszerzeniem wersji dyskowej (w podanym opisie polecenia wersji BASICA są oznaczone <sup>(a)</sup>). Implementacja tego języka została wykonana przez kilka firm. Oprócz tego dla mikrokomputerów typu IBM PC istnieją też inne wersje języka BASIC, czasem znacznie odbiegające od wersji podstawowej.

### *deklaracja tablic*

Tablice mogą mieć do 256 wymiarów i maksymalny rozmiar w każdym wymiarze do 15153 elementów. Wartość pierwszego wskaźnika jest standardowo równa 0, chyba że wcześniej zostanie wykonane polecenie OPTION BASE 1. Istnieje również deklaracja standardowa tablicy według pierwszej użytej zmiennej wskaźnikowej.

### *dlg\_pola*

Wyrażenie całkowite określające liczbę znaków wartości tekstowej lokalizowanej w rekordzie związanym z operacjami wejścia-wyjścia dla pliku o dostępie swobodnym. Długość wszystkich pól rekordu nie może przekraczać → *długości rekordu*.

### *dlg\_rek*

Wyrażenie całkowite określające długość rekordu dla plików o dostępie bezpośrednim, o standardowej wartości równej 128. Długość rekordu nie może przekraczać wartości określonej przy → *wywołaniu systemu Basic*.

*identyfikator*

Identyfikatorem jest ciąg do 40 znaków — złożony z liter, cyfr i kropki — rozpoczynający się od litery. Duże i małe litery są utożsamiane. Standardowo identyfikatory są listowane zawsze z wykorzystaniem dużych liter. Identyfikatory nie mogą być → *słowami kluczowymi*, ale mogą zawierać w sobie słowa kluczowe.

*ekran graficzny*

Monitor ekranowy może być przełączany z trybu funkcjonowania jako → *ekran tekstowy* w tryb funkcjonowania jako → *ekran graficzny* o niskiej lub wysokiej rozdzielczości wybieranej poleceniem:

SCREEN 1 — 320 × 200 punktów w 16 kolorach tła i 4 kolorach atramentu z dwóch → *palet barw*,

SCREEN 1,1 — 320 × 200 punktów w 2 kolorach (ekran dwubarwny),

SCREEN 2 — 640 × 200 punktów w 2 kolorach.

Zakres zmian współrzędnych ekranu jest określony w poziomie *wsp\_x*: 0..319 lub 0..639 oraz w pionie *wsp\_y*: 0..199, ze standardowym położeniem punktu początkowego układu w lewym górnym rogu ekranu. Położenie początku układu może być redefiniowane. Możliwe jest również zdefiniowanie uogólnionych współrzędnych *x* i *y* w zakresie liczb całkowitych. W trybie graficznym można też wyświetlać w ograniczonym stopniu znaki tekstów.

*ekran tekstowy*

Monitor ekranowy znajduje się standardowo w trybie funkcjonowania jako ekran tekstowy lub może być przełączony z trybu → *ekranu graficznego* poleceniem:

SCREEN 0 — 25 wierszy po 40 lub 80 znaków w wierszu, zależnie od ostatnio wykonanego zlecenia WIDTH.

Zakres numeracji wierszy jest określony w przedziale 1..25, a kolumn w przedziale 1..40 lub 1..80. Pozycją początkową kursora wyświetlania jest lewy górny róg ekranu. Do programowego przesuwania kursora wyświetlania w określone miejsce ekranu służy instrukcja LOCATE oraz separatory instrukcji wyprowadzania. Na ekranie tekstowym jest wyświetlana zawartość określonej strony pamięci operacyjnej. Możliwe jest wypełnianie przez program jednej ze stron, gdy inna (lub ta sama) jest aktualnie wyświetlana. Zakres numeracji stron jest równy 0..7 (dla 40 znaków w wierszu) lub 0..3 (dla 80 znaków).

*identyfikacja pliku*

Wyrażenie tekstowe o składni

[*ident\_\_urz*] [*ścieżka*] [*nazwa\_\_pliku*] [*.roz*]

gdzie dopuszczalne są następujące identyfikatory urządzeń:

A:, B:, C:, D: adapter dysku

LPT1: drukarka

SCRN: ekran

PRN: drukarka

CON: klawiatura z ekranem

KYBD: klawiatura

NUL: puste — do nikąd

a → *ścieżka* określa drogę od katalogu głównego do katalogu danego pliku, oraz *nazwa\_pliku* jest identyfikatorem złożonym ze znaków:

A..Z , 0..9 , \$ & # @ ! % ` ( ) — < > ~ { } — \ ^ :

o maksymalnej długości 8 znaków, a *roz* jest 3-znakowym rozszerzeniem precyzującym przeznaczenie pliku.

#### klawiatura

Klawiatura jest ułożona w standardowym układzie QWERTY klawiszy wprowadzających kody znaków małych i dużych liter, cyfr, symboli oraz znaków sterujących. Dodatkowo po lewej stronie klawiatury znajduje się 10 klawiszy funkcyjnych, a po prawej — klawiatura sterująco-numeryczna. Wszystkie klawisze na klawiaturze mają przypisane odrębne numery z zakresu 1..83. Wykorzystanie tych numerów jest związane z możliwością nadania każdemu z klawiszy roli → *klawisz funkcyjnego* (tylko w BASICA).

#### klawisze funkcyjne

Po lewej stronie → *klawiatury* znajduje się 10 klawiszy funkcyjnych. Klawiszom tym są standardowo przypisane teksty — wyświetlane w ostatnim wierszu ekranu po CTRL&T lub listowane poleceniem KEY LIST. W programie, klawiszom tym o numerach z zakresu 1..10 możliwe jest przypisanie poleceniem KEY nowych wartości tekstowych o maksymalnej długości 15 znaków każdy. W wersji BASICA istnieje jeszcze dodatkowych 10 klawiszy funkcyjnych o numerach z zakresu 11..20, przy czym numerom 11..14 są standardowo przypisane klawisze przesuwania kursora odpowiednio — w górę, w lewo, w prawo i w dół. Numerom z zakresu 15..20 można przypisać kod znaku każdego z ponumerowanych klawiszy z modyfikacją jego działaniem → *klawisz sterującego*.

#### klawisze sterujące

Wyróżnione są następujące klawisze o kodach znaków sterujących:



ENTER — zakończenie wprowadzania lub poprawiania linii zawierającej polecenie.



SHIFT — wprowadzenie dużej litery lub symbolu wskazanego równocześnie wciśniętym klawiszem.

CAPS\_LOCK — przełączenie na stałe trybu wprowadzania liter tylko jako dużych — odwołanie przełączenia powtórным wciśnięciem klawisza CAPS\_LOCK.

- CTRL — wprowadzenie znaku sterującego związanego z równocześnie wciśniętym klawiszem.
- ESC — przełączenie na tryb wprowadzania następnie wprowadzanej sekwencji liter jako znaków sterujących.
- ALT — podobne zastosowanie jak CTRL.



BACKSPACE — usunięcie znaku poprzedzającego kursor wprowadzania z przesunięciem kursora w lewo.



TAB — przesunięcie kursora wprowadzania do następnej  $n \cdot 8$  pozycji w prawo; w trybie wstawiania znaków przesunięcie kursora do najbliższej prawej pozycji tabulacji.

- PRTSC — (patrz CTRL&PRTSC).
- NUM\_\_LOCK — przełączenie części sterująco-numerycznej klawiatury w tryb wprowadzania cyfr lub znaków sterujących.
- HOME — przesunięcie kursora w lewy górny róg ekranu.
- PG\_\_UP — (nieużywane).
- PG\_\_DN — (patrz CTRL&PG\_\_DN).
- END — przesunięcie kursora na koniec linii.
- przesuwanie kursora wprowadzania odpowiednio:



w górę,



w lewo,

w prawo,



w dół.

- DEL — usunięcie znaku wskazanego kursorem z przesunięciem w lewo pozostałej części tekstu linii.
- INS — przełączenie na tryb wstawiania następnie wprowadzanych znaków z rozsuwaniem w prawo tekstu linii aż do powtórnego wprowadzenia INS, lub przesunięcia kursora. Pozostałe funkcje sterujące uzyskuje się przez jednoczesne działanie klawiszy znaków sterujących i innych klawiszy:

CTRL&ALT&DEL

— wyzerowanie stanu komputera.

SHIFT&PRTSC

— skopiowanie aktualnej zawartości obrazu ekranu na drukarkę.



**CTRL&PRTSC**

— włączenie kopiowania na drukarkę wszystkich tekstów wyświetlanych na ekranie.

**CTRL&HOME** — czyszczenie ekranu z ustawieniem kursora w lewym górnym rogu ekranu.

**CTRL&J** — przesunięcie kursora na początek następnego wiersza w tej samej linii.

**CTRL&←** — przesunięcie kursora na początek poprzedniego słowa.

**CTRL&→** — przesunięcie kursora na następne słowo.

**CTRL&END** — usunięcie tekstu do końca linii.

**CTRL&PG\_DN** — usunięcie tekstu do końca ekranu.

**CTRL&BREAK**

— powrót do trybu wprowadzania nowego zlecenia.

**CTRL&NUM\_LOCK**

— wstrzymanie działania programu — do chwili naciśnięcia dowolnego klawisza.

**CTRL&T** — włączenie wyświetlania tekstów przypisanych klawiszom funkcyjnym i następnie po CTRL&T wygaszenie tych tekstów.

*kursor*

Wartość logiczna określająca widoczność kursora: prawda — kursor widoczny, fałsz — niewidoczny.

*linia programu*

W linii może być do 255 znaków. Zakres numeracji linii wynosi 1..65535.

*nr\_palety*

Wyrażenie całkowite o wartości (0..255) modulo 2, wskazujące jedną z dwóch → *palet barw* dla koloru atramentu w trybie → *ekranu graficznego* o niskiej rozdzielczości.

*nr\_rek*

Wyrażenie całkowite o wartości z zakresu 1..32767 (lub 1..16777215 w BASICA) wskazujące numer rekordu w pliku o dostępie swobodnym.

*nr\_strony\_akt nr\_strony\_wid*

Wyrażenia całkowite o wartościach z zakresu 0..7 (dla 40 znaków w wierszu → *ekranu tekstowego*) lub 0..3 (dla 80 znaków) wskazujące, która ze stron ma być aktualnie wypełniana treścią, a która wyświetlana.

*numer logiczny strumienia*

Wyrażenie całkowite o wartości z zakresu 0..3 (lub 0..15, zależnie od parametrów → *wywołania systemu Basic*), wskazujące numer strumienia. Numer logiczny strumienia, wykorzystywany w instrukcjach wprowadzania i wyprowadzania, jest wiązany z rzeczywistym plikiem w instrukcji otwarcia pliku OPEN.

*operatory*

- arytmetyczne:  $\uparrow$  - \* / \ MOD + -
- relacji : = < = < <> > > =
- logiczne : NOT AND OR XOR EQV
- tekstowe : + oraz relacji

Operatory logiczne funkcjonują też na bitach rozwinięcia dwójkowego wartości liczbowych całkowitych.

*opis zakresu numeracji linii*

Zakres numeracji linii określa fragment programu, który ma być poddany działaniu zlecenia i może przyjmować postać:

- pusty — cały tekst programu,
- .
- nr\_linii* — tylko linia o podanym numerze,
- nr\_linii* - — fragment od danej linii do końca tekstu programu,
- *nr\_linii* — fragment od pierwszej linii programu do podanej włącznie,
- nr\_linii* - *nr\_linii* — fragment programu ograniczony liniami o podanych numerach.

*paleta barw*

Dla monitora kolorowego jest dostępnych 16 kolorów:

0 16 czarny	8 24 szary
1 17 niebieski	9 25 jasnoniebieski
2 18 zielony	10 26 jasnozielony
3 19 siny	11 27 jasnosiny
4 20 czerwony	12 28 jasnoczerwony
5 21 karminowy	13 29 jasnokarminowy
6 22 brązowy	14 30 żółty
7 23 biały	15 31 intensywnie biały

Numery z zakresu 16..31 oznaczają znaki mrugające w danym kolorze. Podane kolory dotyczą jedynie tła obrazu.

Kolory atramentu mogą być wybierane z jednej z dwóch palet o podanym  $\rightarrow$  *nr\_palety*:

paleta 0 (parzysta)	paleta 1 (nieparzysta)
0 kolor tła	0 kolor tła
1 zielony	1 siny
2 czerwony	2 karminowy
3 żółto-brązowy	3 biały

Dla monitora monochromatycznego i dla drukarki podane numery kolorów odpowiadają następującemu zobrazowaniu:

atrament	tło
0 czarny	czarne
1 biały-podkreślony	czarne
2..6 biały	czarne
7 czarny	białe

8..15 intensywność jaskrawość

16..31 mruganie

#### *poprawianie linii programu*

Pojedyncze linie programu w trakcie wprowadzania lub po sprawdzeniu poleceniem EDIT oraz z wykrytym podczas wykonywania błędem składniowym (*Syntax Error*), lub wylistowane na ekranie poleceniem LIST mogą być poprawiane następująco:

- przesunąć kursor wprowadzania w błędne miejsce linii, korzystając z klawiszy ↑ ← → ↓,
- usunąć znaki z tekstu linii klawiszami ◀ lub DEL,
- wymienić błędne znaki na nowe w treści linii wprowadzając je w miejsce wskazane kursorem,
- wpisać nowe znaki w tekst linii w miejsca wskazane kursorem po przełączeniu w tryb wprowadzania klawiszem INS aż do ponownego użycia klawisza INS,
- zakończyć poprawianie linii klawiszem ENTER.

#### *postać zapisu pliku*

W pamięci kasetowej oraz dyskowej mogą być zapisywane pliki następujących typów:

- B — z programem w Basicu w postaci upakowanej binarnie,
- P — z programem w Basicu w postaci binarnej z protekcją odczytu,
- A — z programem w kodzie ASCII,
- M — z obrazem pamięci operacyjnej w postaci binarnej,
- D — z danymi dla programu.

Litery A i P są używane do określenia typu pliku przy jego składowaniu. Pusty typ pliku odpowiada literze B.

#### *rodzaj pliku*

Wykorzystując pamięć dyskową można współpracować z plikami danych i wyników w następujących trybach:

- I INPUT — sekwencyjny plik wejściowy dla odczytu danych,
- O OUTPUT — sekwencyjny plik wyjściowy dla zapisu wyników,
- APPEND — kontynuacja po ostatnim zapisie sekwencyjnego pliku wyjściowego dla zapisu wyników,
- R — plik o dostępie swobodnym.

Podane oznaczenia trybu otwarcia pliku są używane w poleceniu OPEN.

#### *rozm\_\_stosu*

Wyrażenie całkowite deklarujące żądaną wielkość stosu na wykonywanie programu. Standardowy rozmiar stosu wynosi 512 bajtów.

#### *słowo kluczowe*

Pełna wersja języka zawiera 172 słowa kluczowe, uwzględniając również dostępne z programu zlecenia systemu DOS. Słowo kluczowe w tekście



punktach oraz drugi koniec linii we *wsp*\_\_*x*, *wsp*\_\_*y*. Instrukcje te mogą być poprzedzone prefiksem definicji parametrów rysunku:

B — przesunąć kursor bez rysowania,

N — narysuj i wróć,

An — obróć rysunek o 90°, 180° lub 270° gdy  $n = 1, 2, 3$ .

#### *typ zmiennej i funkcji*

Wyróżnione są cztery typy zmiennych i funkcji deklarowanych pierwszą literą identyfikatora określoną w poleceniu DEFxxx, lub też znacznikiem typu:

% — liczbowego całkowitego,

! — liczbowego rzeczywistego pojedynczej precyzji (brak znacznika określa ten typ z założenia),

# — liczbowego rzeczywistego podwójnej precyzji,

\$ — tekstowego.

#### *wartość liczbowa*

Dopuszczalne są wartości liczbowe typu:

- całkowitego z zakresu  $-32768..32767$  zapisane w postaci *cyfra*\_\_10... dziesiętnej (zakończony znacznikiem typu %),
- &O *cyfra*\_\_8... ósemkowej,
- &cyfra\_\_8... ósemkowej,
- &H *cyfra*\_\_16... szesnastkowej.

W niektórych zastosowaniach wartość całkowita jest traktowana jako ciąg 16 bitów w zapisie uzupełnień do dwóch:

- rzeczywistego z zakresu  $-10^{38}..-10^{-38}, 0, 10^{-38}..10^{38}$
- — pojedynczej precyzji z dokładnością do 7 cyfr znaczących w postaci liczby całkowitej lub ułamkowej ewentualnie zakończonej znacznikiem typu ! i wykładniczej z wykładnikiem oznaczonym literą E,
- — podwójnej precyzji z dokładnością do 17 cyfr znaczących w postaci liczby całkowitej lub ułamkowej zakończonej znacznikiem typu # i wykładniczej z wykładnikiem oznaczonym literą D.

#### *wartość logiczna*

Wartość logiczna prawda (nie zero), fałsz (0). → *Operatory logiczne* działają na 16 bitach pełnego rozwinięcia → *wartości liczbowej* całkowitej.

#### *wartość tekstowa*

Teksty mogą zawierać od 0 do 255 znaków o dowolnej wartości kodu znaku z zakresu 1..255.

*wyr\_aryt\_poj\_prec*      *wyr\_aryt\_podw\_prec*

Wyrażenia arytmetyczne o wartości rzeczywistej pojedynczej lub podwójnej precyzji przeznaczonej do konwersji w 4 lub odpowiednio 8 bajtowy tekst.

*wyr\_tekst\_2bajt*      *wyr\_tekst\_4bajt*      *wyr\_tekst\_8bajt*

Wyrażenia tekstowe przechowujące odpowiednio na 2, 4 lub 8 bajtach wartości przeznaczone do konwersji odpowiednio w wartość całkowitą lub pojedynczej oraz podwójnej precyzji.

*wyr\_tekst\_czasu*      *wyr\_tekst\_daty*

Wyrażenie tekstowe o określonej postaci reprezentującej bieżący czas "hh.mm.ss" oraz aktualną datę "mm-dd-yy".

*wywołanie systemu Basic*

**Cassette BASIC** zgłasza się bezpośrednio po włączeniu mikrokomputera — jeżeli w żadnym z adapterów nie ma dyskietki.

**Disk BASIC** musi być załadowany pod kontrolą systemu CP/M-86 lub DOS przez wprowadzenie polecenia BASIC.

**Advanced BASIC** ładowany analogicznie przez wprowadzenie polecenia BASICA<sup>1)</sup>. Polecenie to może posiadać kilka parametrów instalacyjnych takich jak:

- |                          |  |
|--------------------------|--|
| <i>spec_pliku</i>        | — identyfikator pliku, z którego zostanie natchmianst załadowany i uruchomiony program zapisany w języku Basic;                |
| < <i>std_we</i>          | — identyfikator urządzenia lub pliku, które będzie standardowym, zamiast klawiatury, urządzeniem wejściowym dla systemu Basic; |
| > <i>std_wy</i>          | — identyfikator standardowego urządzenia lub pliku wyjściowego;  |
| /F: <i>liczba_plików</i> | — maksymalna liczba plików, które mogą być równocześnie otwarte (standardowo 3);   |
| /S: <i>dłg_rek</i>       | — maksymalna dopuszczalna długość rekordu dla plików o dostępie swobodnym;   |
| /C: <i>dłg_bufor</i>     | — przydzielenie określonej długości bufora dla odbioru informacji z łącza komunikacyjnego RS 232;                              |
| /M: <i>max_obszar</i>    | — określenie maksymalnej wartości zakresu obszaru pamięci przeznaczonej na potrzeby systemu Basic (max: 65536).                |
| /D                       | — dołączenie procedur obliczania funkcji w podwójnej precyzji.   |

<sup>1)</sup> Odpowiednikiem BASICA jest niezależny od istnienia w ROM podprogramów Basicu — program tłumacza zapisany w pliku gwbasic.exe.

*zaokrąglenia*

Wartość rzeczywista jest zaokrąglana, jeżeli jest to konieczne, do najbliższej wartości całkowitej.

**Wykaz poleceń***Edycja programu*

```

NEW
AUTO [nr_linii[,inkrement]]
<CTRL&BREAK>                               'wstrzymanie AUTO
LIST [zakres_linii] [,spec_pliku]
LLIST [zakres_linii]
DELETE zakres_linii
EDIT [.]
EDIT nr_linii
RENUM [[nr_linii_nowy] [,nr_linii_stary] [,inkrement]]]

```

*Komentarze*

```

REM dowolny_tekst
'dowolny_tekst

```

*Uruchomienie i śledzenie programu*

```

@ ENVIRONS$(wyr_tekst)
@ ENVIRONS$(wyr_calk)
RUN [nr_linii]
RUN spec_pliku [,R]
CONT
TRON
TROFF
SYSTEM
@ SHELL [wyr_tekst]
@ ENVIRON wyr_tekst

```

*Ładowanie, składowanie i wiązanie programów*

```

SAVE spec_pliku [,typ_pliku]
LOAD spec_pliku [,R]
MERGE spec_pliku
CHAIN [MERGE] spec_pliku [[,wyr_nr_linii] [,ALL]
                                [,DELETE zakres_linii]]
COMMON zmienna [,zmienna]...

```

*Definicje i deklaracje*

```

DEFINT zakres_liter
DEFSNG zakres_liter
DEFDBL zakres_liter
DEFSTR zakres_liter
DIM nazwa_tabl (rozm[,rozm]...) [,nazwa_tabl(rozmiary)]...

```

OPTION BASE *par01*

ERASE *nazwa\_\_tabl* [,*nazwa\_\_tabl*]...

DEF FN*nazwa\_\_fun* [(*par* [,*par*]...)] = *wyrażenie*

#### Obliczenia arytmetyczne

FN*nazwa\_\_fun\_\_aryt* [(*arg*[,*arg*]...)]

ABS (*wyr\_\_aryt*)

ATN (*wyr\_\_aryt*)

COS (*wyr\_\_aryt*)

EXP (*wyr\_\_aryt*)

FIX (*wyr\_\_aryt*)

INT (*wyr\_\_aryt*)

LOG (*wyr\_\_aryt*)

SGN (*wyr\_\_aryt*)

SIN (*wyr\_\_aryt*)

SQR (*wyr\_\_aryt*)

TAN (*wyr\_\_aryt*)

CDBL (*wyr\_\_aryt*)

CINT (*wyr\_\_aryt*)

CSNG (*wyr\_\_aryt*)

VAL (*wyr\_\_tekst*)

RND (*wyr\_\_aryt*)

@ TIMER

[LET] *zm\_\_aryt* = *wyr\_\_aryt*

SWAP *zm\_\_aryt*,*zm\_\_aryt*

RANDOMIZE [*wyr\_\_calk*]

@ RANDOMIZE TIMER

#### Przetwarzanie tekstów

FN *nazwa\_\_fun\_\_tekst* [(*arg*[,*arg*]...)]

ASC (*wyr\_\_tekst*)

CHR\$ (*kod\_\_znaku*)

DATE\$

HEX\$ (*wyr\_\_calk*)

INSTR ([*nr\_\_pozycji*,] *wyr\_\_tekst*,*wyr\_\_tekst*)

LEFT (*wyr\_\_tekst*,*dlg\_\_tekstu*)

LEN (*wyr\_\_tekst*)

MID\$ (*wyr\_\_tekst*,*nr\_\_pozycji* [,*dlg\_\_tekstu*])

OCT\$ (*wyr\_\_calk*)

RIGHT\$ (*wyr\_\_tekst*,*dlg\_\_tekstu*)

SPACE\$ (*dl\_\_tekstu*)

STR\$ (*wyr\_\_aryt*)

STRING\$ (*dlg\_\_tekstu*,*kod\_\_znaku*)

STRING\$ (*dlg\_\_tekstu*,*wyr\_\_tekst*)

TIME\$



```
[LET] zm__tekst = wyr__tekst
MID$ (zm__tekst, nr__pozycji [,dlg__tekstu]) = wyr__tekst
DATE$ = wyr__tekst__daty
TIME$ = wyr__tekst__czasu
SWAP zm__tekst,zm__tekst
```

#### *Sterowanie programem*

```
END
STOP
GOTO nr__linii
IF wyr__log [,] GOTO nr__linii [ELSE nr__linii__lub__instrukcje]
IF wyr__log [,] THEN nr__linii__lub__instrukcje
                                [ELSE nr__linii__lub__instrukcje]
FOR zm__ster = wart__pocz TO wart__końc [STEP wart__kroku]
NEXT zm__ster [,zm__ster]...
WHILE wyr__log
WEND
GOSUB nr__linii
RETURN [nr__linii]
ON wyr__calk GOTO nr__linii [,nr__linii]...
ON wyr__calk GOSUB nr__linii [,nr__linii]...
@ ON TIMER (okres) GOSUB nr__linii
```

#### *Wprowadzanie danych*

```
INKEY$
INPUT$ (dlg__tekstu)
INPUT [;] [”tekst__zachęty”;] zmienna [,zmienna]...
LINE INPUT [;] [”tekst__zachęty”;] zm__tekst
DATA stała [,stała]...
READ zmienna [,zmienna]...
RESTORE [nr__linii]
KEY nr__klawisza, wyr__tekst
KEY LIST
KEY ON
KEY OFF
@ KEY (nr__klawisza) ON
@ KEY (nr__klawisza) OFF
@ KEY (nr__klawisza) STOP
@ KEY nr__klawisza, CHR$ (kod__ster) + CHR$ (nr__klawisza)
@ ON KEY (nr__klawisza) GOSUB nr__linii
```

#### *Wyrowadzanie wyników*

```
CSRLIN
POS(0)
```





```

PEEK (wyr__adr)
USR [cyfra] (wyr__calk)
VARPTR (zmienna)
VARPTR (#nr__str)
VARPTR $ (zmienna)
CLEAR [, [max__adres] [, rozm__stosu]]
OUT nr__portu, wyr__calk
WAIT nr__portu, wyr__calk [, wyr__calk]
DEF USR [cyfra] = wyr__calk
DEFSEG [= wyr__adr] .
POKE wyr__adr, wyr__calk
CALL zmienna [(arg[, arg]...)]
BLOAD spec__pliku [, wyr__adr]
BSAVE spec__pliku, wyr__adr, wyr__adr

```

#### Generowanie dźwięku

```

■ PLAY(0)
BEEP
SOUND częst,okres
@ PLAY "tekst__opisu__muzyki"
@ ON PLAY (wyr__calk) GOSUB nr__linii

```

#### Współpraca z innymi urządzeniami

```

■ PEN(par 0..9)
STICK (par01234)
STRIG (par0123)
PEN ON
PEN OFF
@ ON PEN GOSUB nr__linii
STRIG ON
STRIG OFF
@ STRIG (par02) ON
@ STRIG (par02) OFF
STRIG (par02) STOP
@ ON STRIG (par02) GOSUB nr__linii
@ OPEN "COMpar1234: parametry" AS[#]nr__str [LEN = dlg__rek]
WIDTH wyr__tekst, szer__wiersza
@ COM (par1234) ON
@ COM (par1234) OFF
@ COM (par1234) STOP
@ ON COM (par1234) GOSUB nr__linii

```

## Wykaz komunikatów błędów

- 73 **Advanced Feature**  
Użyto elementów języka dostępnych tylko w jego rozszerzonej wersji.
- 54 **Bad File Mode**  
Próba wykonania operacji niezgodnej z postacią lub rodzajem pliku.
- 64 **Bad File Name**  
Błędny zapis specyfikacji pliku.
- 52 **Bad File Number**  
Błędna wartość numeru logicznego strumienia nie związana z otwartym plikiem lub spoza zakresu.
- 63 **Bad Record Number**  
Błędna wartość numeru rekordu dla operacji na pliku o dostępie swobodnym.
- 17 **Can't Continue**  
Próba kontynuacji programu, który został przerwany w wyniku wykrycia sytuacji błędnej, został zmodyfikowany lub nie istnieje w pamięci.
- 69 **Communication Buffer Overflow**  
Przepełnienie buforu komunikacyjnego spowodowane za małą jego długością, bądź brakiem odpowiedniego protokołu współpracy lub za dużą szybkością transferu przesyłania informacji.
- 25 **Device Fault**  
Awaria sprzętowa drukarki.
- 57 **Device I/O Error**  
Błąd fatalny wykryty podczas operacji we-wy.
- 24 **Device Timeout**  
Brak informacji z urządzenia we-wy w określonym przedziale czasu.
- 68 **Device Unavailable**  
Próba otwarcia pliku na nieistniejącym nośniku lub brak w systemie podanego urządzenia.
- 66 **Direct Statement in File**  
Podczas ładowania programu wykryto linię bez numeru traktowaną jako zlecenie.
- 61 **Disk Full**  
Brak miejsca na nośniku pamięci dyskowej.
- 72 **Disk Media Error**  
Błąd nośnika dyskietki. Po skopiowaniu wszystkich plików na inną dyskietkę — wadliwą dyskietkę należy przeformatować.
- 70 **Disk Write Protect**  
Próba zapisu na dyskietkę z protekcją zapisu.

- 11 Division By Zero  
Wykryto próbę dzielenia przez zero lub podnoszenia wartości zero do potęgi ujemnej. Wynikiem wykonania tych operacji jest wartość maksymalna z odpowiednim znakiem oraz kontynuacja wykonywania programu.
- 10 Duplicate Definition  
Próba powtórnej deklaracji tej samej tablicy.
- 50 Field Overflow  
Próba deklaracji rekordu o długości przekraczającej ustaloną przez parametr instalacji przy wywoływaniu systemu Basic.
- 58 File Already Exists  
Próba nazwania pliku nazwą już istniejącą w danym katalogu.
- 55 File Already Open  
Próba ponownego otwarcia pliku już otwartego lub próba jego zlikwidowania.
- 53 File Not Found  
Próba odwołania się do pliku, który nie istnieje.
- 26 FOR Without NEXT  
W programie brak instrukcji NEXT domykającej instrukcję cyklu wyliczanego rozpoczynającą się instrukcją FOR.
- 12 Illegal Direct  
Wprowadzone polecenie nie może być wykonane jako zlecenie.
- 5 Illegal Function Call  
Błędny argument wywoływanej funkcji — przykładowo: ujemna lub spoza zakresu wartość wskaźnika tablicy, ujemna wartość argumentu funkcji SQR lub dla LOG również zerowa oraz ujemna wartość mantysy przy podnoszeniu do całkowitej potęgi, itp.
- Incorrect DOS Version  
Potrzebna inna wersja systemu DOS.
- 62 Input Past End  
Próba odczytu z pliku pustego lub odczytanego już do znacznika końca.
- 51 Internal Error  
Błąd systemu Basic (tylko autorzy interpretatora mogą go poprawić).
- 23 Line Buffer Overflow  
Próba wprowadzenia linii o zbyt dużej liczbie znaków.
- 22 Missing Operand  
Nieprawidłowa składnia wyrażenia.
- 1 NEXT Without FOR  
Zmienna sterująca wymieniona w instrukcji NEXT nie występuje w żadnej odpowiedniej instrukcji początku cyklu FOR.
- 19 No RESUME

- Brak instrukcji RESUME kończącej podprogram obsługi sytuacji błędnych.
- 4 Out of DATA  
Brak danych dla instrukcji READ.
  - 7 Out of Memory  
Tekst programu i obszar pamięci potrzebny na jego zmienne oraz wykonanie nie mieści się w dostępnej pamięci operacyjnej.
  - 27 Out of paper  
Brak papieru w drukarce lub nie jest ona włączona.
  - 14 Out of String Space  
Brak miejsca na zapisanie wartości tekstowych.
  - 6 Overflow  
Wynik operacji arytmetycznych przekracza dopuszczalny zakres wartości liczbowych.
  - 75 Path/file access error  
Próba użycia ścieżki dostępu do niedostępnego pliku.
  - 76 Path not found  
Nierozpoznawalna ścieżka w katalogu plików.
  - 74 Rename across disks  
Próba przemianowania pliku istniejącego na innej niż określona dyskietce.
  - 20 RESUME Without Error  
Próba wykonania instrukcji RESUME poza podprogramem obsługi sytuacji błędnej.
  - 3 RETURN Without GOSUB  
Próba wykonania instrukcji RETURN, dla której nie było wykonania instrukcji wywołania podprogramu RESUME.
  - 16 String Formula Too Complex  
Zbyt skomplikowane obliczenia w wyrażeniu tekstowym.
  - 15 String Too Long  
Próba utworzenia tekstu o długości większej niż 255 znaków.
  - 9 Subscript Out of Range  
Nieprawidłowa liczba wskaźników lub wartość jednego z nich przekracza zakresy określone w trakcie deklaracji tablicy.
  - 2 Syntax Error  
Niezrozumiały zapis polecenia — błędy w składni.
  - 67 Too Many Files  
Próba otwarcia współpracy z plikiem, gdy wyczerpano już liczbę jednocześnie otwartych plików.
  - 13 Type Mismatch  
Niezgodność typów wartości między argumentami a odpowiadającymi im parametrami.

- 8 Undefined Line Number  
Brak linii o danym numerze.
- 18 Undefined User Function  
Brak definicji funkcji USR
- 21 Unprintable Error  
Brak tekstu opisującego sytuację wyjątkową o danym numerze błędu.
- 30 WEND Without WHILE  
Wykonywanie instrukcji WEND, dla której nie ma odpowiadającej jej instrukcji WHILE.
- 29 WHILE Without WEND  
W programie brak instrukcji WEND domykającej cykl z warunkiem rozpoczęty instrukcją WHILE.



## 5. ZX Spectrum (Plus)

Mikrokomputer osobisty ZX Spectrum i równoważny mu ZX Spectrum Plus jest standardowo wyposażony w interpretator języka Basic zawarty w pamięci ROM. Ta wersja języka Basic jest równoważna ograniczonej wersji języka BASIC Microsoft ze znacznym uzupełnieniem w zakresie grafiki. Omówiona tutaj wersja konfiguracji mikrokomputera współpracuje jedynie z pamięcią kasetową.

Istotnym problemem przy korzystaniu z tego mikrokomputera jest wprowadzanie tekstu programu. Każdy z klawiszy może działać w jednym z pięciu trybów wprowadzając słowa kluczowe języka Basic, pojedyncze znaki i symbole lub znak sterujący. Różnica między ZX Spectrum a ZX Spectrum Plus polega na innym kształcie klawiszy klawiatury uzupełnionej o osobne klawisze najczęściej używanych znaków sterujących.

### *deklaracja tablic*

Tablice mogą mieć dowolną liczbę wymiarów i rozmiarów z minimalną wartością wskaźnika równą 1. W tablicach typu tekstowego ostatni wymiar określa maksymalną długość tekstu przechowywanego znak po znaku w kolejnych elementach tablicy. Możliwa jest powtórna deklaracja tablicy o tym samym identyfikatorze.

### *identyfikatory*

Identyfikatorem jest ciąg znaków — złożony z liter i cyfr — rozpoczynający się od litery. Małe i duże litery są utożsamiane. Zależnie od → *typu zmiennej lub funkcji* identyfikatory:

- zmiennych liczbowych mogą być dowolnej długości,
- zmiennych tekstowych mogą być jedynie jednoliterowe,
- zmiennych sterujących muszą być jednoliterowe,
- tablic liczbowych i tekstowych mogą być jedynie jednoliterowe,
- funkcji mogą być tylko jednoliterowe.

Identyczne identyfikatory zmiennych, tablic i funkcji oznaczają różne obiekty. Słowa kluczowe mogą być częścią identyfikatora. → *Słowa kluczowe* wprowadzane jako ciąg liter nie są rozpoznawane przez interpretator.

#### *ekran graficzny*

Ekran graficzny jest utożsamiony z → *ekranem tekstowym* i funkcjonuje w trybie graficznym dla  $256 \times 176$  punktów w 2 kolorach (tła znaku, do którego należy i koloru atramentu) spośród 8 kolorów z → *palety barw*. Zakres zmian współrzędnych jest określony w poziomie *wsp\_x*: 0..255 oraz w pionie *wsp\_y*: 0..175, ze standardowym położeniem punktu początkowego układu w lewym dolnym rogu 22 wiersza ekranu.

#### *ekran tekstowy*

Ekran tekstowy jest utożsamiony z → *ekranem graficznym* i funkcjonuje w trybie tekstowym dla 24 wierszy i 32 znaków w wierszu. Pierwsze 22 wiersze od góry ekranu są przeznaczone do wyświetlania znaków tekstów wyprowadzanych. Pozycja wyświetlanego znaku jest określana w zakresie *nr\_wiersza*: 0..21 i *nr\_kol*: 0..31. Początkową pozycją kursora wyświetlania jest lewy górny róg ekranu. Możliwe jest programowe lokalizowanie kursora wyświetlania na ekranie korzystające z frazy AT w zakresie zmienności numerów wierszy i kolumn. Dwa ostatnie wiersze są przeznaczone do wyświetlania tekstu wprowadzanego. Obszar ten może być od dołu rozszerzany aż do zajęcia 22 wierszy ekranu. Każdy znak na ekranie może być jednego z 8 kolorów tła oraz jednego z 8 kolorów atramentu wybranych z palety barw. Dodatkowo każdy znak może mrugać, być rozjaśniony oraz wyświetlony w postaci negatywu.

#### *identyfikacja pliku*

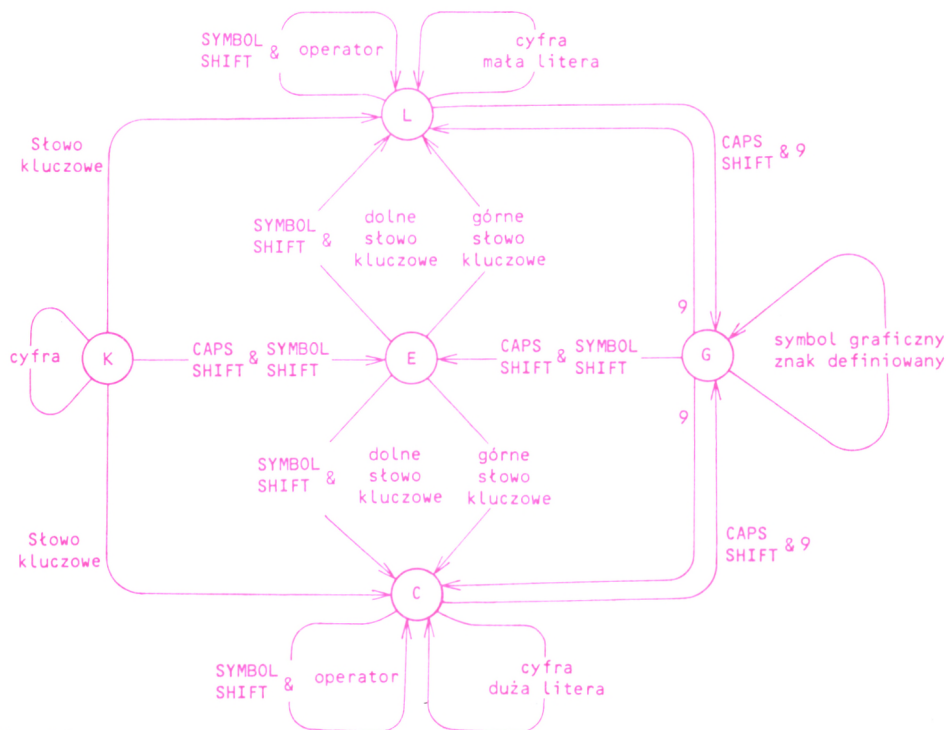
Pliki w pamięci kasetowej są identyfikowane nazwą pliku będącą tekstem złożonym z dowolnych znaków o maksymalnej długości nie przekraczającej 10 znaków. Przy ładowaniu treści pliku nazwa pusta pliku "" dotyczy każdego pierwszego pliku odczytywanego z kasy.

#### *klawiatura*

Klawiatura jest ułożona w standardowym układzie QWERTY klawiszy, liter, cyfr i symboli uzupełnionych klawiszami sterującymi zmieniającymi funkcję pozostałych klawiszy poprzez wybór odpowiedniego trybu wprowadzania. W ZX Spectrum Plus zachowano standard funkcjonowania klawiatury oraz klawisze sterujące ułatwiające zmianę trybu wprowadzania i klawisze symboli , . " ; co przyspiesza wprowadzanie treści programu. O aktualnym trybie wprowadzania znaków przez poszczególne klawisze klawiatury informuje postać kursora wprowadzania na ekranie:

- K** — tryb wprowadzania cyfr numeru linii oraz słów kluczowych rozpoczynających treść polecenia — wskazanych klawiszami liter; tryb ten pojawia się na początku nowo wprowadzanej linii, po wprowadzeniu słowa THEN lub symbolu dwukropka,
- L** — tryb wprowadzania małych liter oraz cyfr,
- C** — tryb wprowadzania dużych liter oraz cyfr,
- G** — tryb wprowadzania symboli graficznych wskazanych na klawiszach cyfr oraz znaków definiowanych opisem znaku związanym z klawiszami liter w zakresie 'A'..'U',
- E** — tryb rozszerzający funkcje wprowadzania symboli i słów kluczowych oznaczonych nad i pod klawiszami.

Przełączanie między trybami ilustruje rys. 5.1, a wykorzystywanie funkcji klawisza rys. 5.2.



Rys. 5.1

### klawisze sterujące

Wyróżnione są następujące klawisze o kodach znaków sterujących:

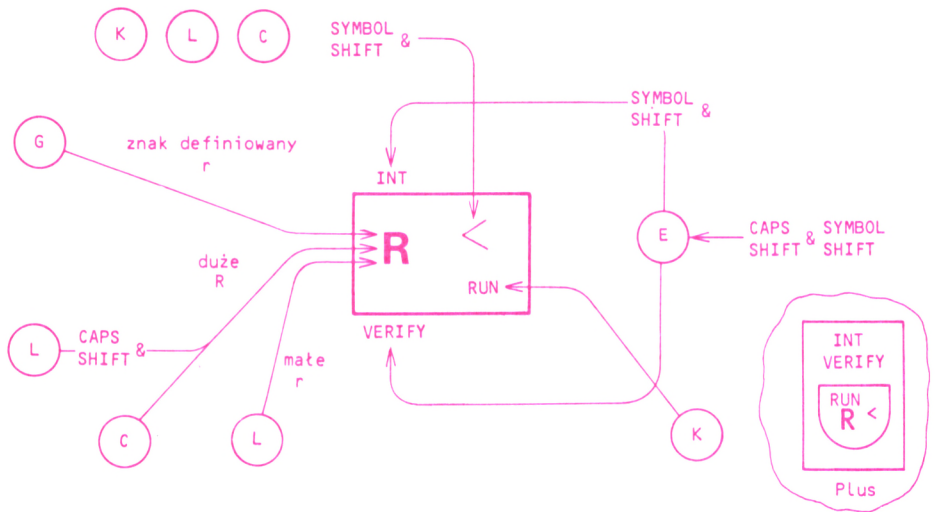
- ENTER** — zakończenie wprowadzania lub poprawiania treści polecenia; zgłoszenie żądania wylistowania fragmentu programu poprzednio listowanego.

**CAPS\_\_SHIFT**

- wprowadzenie dużej litery, funkcji sterującej, negatywu znaku lub symbolu graficznego — wskazanego równocześnie wciśniętym klawiszem.

**SYMBOL\_\_SHIFT**

- wprowadzenie operatora oznaczonego na czerwono na klawiszu równocześnie wciśniętym; w trybie rozszerzającym funkcje wprowadzania — symbolu oznaczonego na czerwono pod klawiszem równocześnie wciśniętym.

**Rys. 5.2**

Pozostałe funkcje sterujące uzyskuje się poprzez jednoczesne działanie klawiszy sterujących i innych klawiszy. W ZX Spectrum Plus wyodrębniono niektóre funkcje sterujące umożliwiając ich wywołanie pojedynczym klawiszem dalej oznaczonym przez <sup>+</sup>:

**CAPS\_\_SHIFT&1****EDIT<sup>+</sup>**

- sprowadzenie linii programu wskazanej na ekranie znacznikiem > do edycji; usunięcie treści aktualnie wprowadzanej lub poprawianej linii.

**CAPS\_\_SHIFT&2****CAPS\_\_LOCK<sup>+</sup>**

- przełączenie w tryb C lub w trybie C w tryb L.

CAPS\_SHIFT&3  
TRUE\_VIDEO<sup>+</sup>

— przełączenie na wprowadzanie kolejnych znaków w formie pozytywu.

CAPS\_SHIFT&4  
INV\_VIDEO<sup>+</sup>

— przełączenie na wprowadzanie kolejnych znaków w formie negatywu (inwersji znaku).

 CAPS\_SHIFT&5

— przesunięcie kursora we wprowadzanej lub poprawianej linii w lewo.

 CAPS\_SHIFT&6

— przesunięcie znacznika linii '>' w kierunku zmniejszających się numerów linii.

 CAPS\_SHIFT&7

— przesunięcie znacznika linii '>' w kierunku zwiększających się numerów linii.

 CAPS\_SHIFT&8

— przesunięcie kursora we wprowadzanej lub poprawianej linii w prawo.

CAPS\_SHIFT&9  
GRAPHICS<sup>+</sup>

— przełączenie w tryb graficzny wprowadzania symboli graficznych lub znaków definiowanych oraz w trybie graficznym przełączenie z powrotem w tryb L lub C.

CAPS\_SHIFT&0  
DELETE<sup>+</sup>

— usunięcie znaku we wprowadzanej lub poprawianej linii z przesunięciem pozostałej części tekstu w lewo.

CAPS\_SHIFT&SPACE

BREAK<sup>+</sup>

— przerwanie wykonywania zlecenia lub programu.

CAPS\_SHIFT&SYMBOL\_SHIFT

EXTEND\_MODE<sup>+</sup>

— przełączenie w tryb E rozszerzający funkcjonalnie klawiaturę dla wprowadzania dodatkowych symboli oraz ewentualnie powrót z tego trybu w tryb L lub C.

W trybie rozszerzającym E możliwe jest wprowadzenie następujących funkcji sterujących lub symboli specjalnych:

0..7 — wprowadzenie nowego koloru tła opisanego nad danym klawiszem.

- 8 — wprowadzenie zwykłej jasności dla kolejnych znaków.
- 9 — wprowadzenie znaku rozjaśniającego tło dla kolejno wprowadzanych znaków.
- CAPS\_\_SHIFT&0..7  
— wprowadzenie nowego koloru atramentu.
- CAPS\_\_SHIFT&8  
— wyłączenie mrugania następnych znaków.
- CAPS\_\_SHIFT&9  
— włączenie mrugania dla następnie wprowadzanych znaków.
- litera*  
— wprowadzenie symbolu oznaczonego nad klawiszem litery.
- SYMBOL\_\_SHIFT&*litera*  
— wprowadzenie symbolu oznaczonego pod klawiszem litery.
- SYMBOL\_\_SHIFT&*cyfra*  
— wprowadzenie symbolu oznaczonego pod klawiszem cyfry.

### *linia programu*

W linii może być teoretycznie dowolna liczba znaków — praktycznie do 255, gdyż wprowadzanie następnych jest coraz wolniejsze. Zakres numeracji linii wynosi 1..9999.

### *operatory*

- arytmetyczne:  $\uparrow$  - \* / + - BIN AND OR
- relacji : = < <= >= > <>
- logiczne : NOT OR AND
- tekstowe : + (*nr\_pozycji* TO *nr\_pozycji*) AND oraz relacji

Operator jednoargumentowy BIN z argumentem *wart\_binarna* dokonuje konwersji z postaci binarnej (dwójkowej) w wartość liczbową całkowitą. Operator TO wycina fragment tekstu zawarty w tekstowej zmiennej wskaźnikowej. Operatory AND i OR mają dodatkowe zastosowanie dla argumentów liczbowych i tekstowych:

- a AND b    jeżeli b <> 0 to a, w przeciwnym razie 0,
- t\$ AND b    jeżeli b <> 0 to t\$, w przeciwnym razie "",
- a    OR b    jeżeli b <> 0 to 1, w przeciwnym razie a.

### *opis zakresu numeracji linii*

Zakres numeracji linii określający fragment programu dotyczy jedynie polecenia LIST i przyjmuje postacie:

pusty — cała treść programu,  
 nr\_linii — fragment od danej linii do końca programu.

#### opis znaku

Każdy znak przypisany literze z zakresu 'A'..'U' wprowadzanej w trybie graficznym G może mieć zdefiniowany nowy kształt przez podanie wartości 8 kolejnych bajtów jego rastra. Kolejne wartości bajtów muszą być umieszczone w pamięci przypisanej znakowi z wykorzystaniem funkcji `USR t$ + i`, gdzie `t$ = "A".."U"` oraz `i = 0..7`.

#### paleta barw

Dla ekranu kolorowego (TV systemu PAL) jest dostępnych osiem kolorów:



0 czarny	2 czerwony	4 zielony	6 żółty
1 niebieski	3 karmazynowy	5 siny	7 biały

Numery kolorów są wykorzystywane w definicji koloru ramki, tła papieru i atramentu.

#### poprawianie linii programu

Pojedyncza linia programu w trakcie wprowadzania lub sprowadzona do ostatniego wiersza ekranu klawiszem `EDIT+` (klawisz ten sprowadza linię wskazaną znacznikiem `>`; znacznik ten może być przesuwany w

kolejne linie za pomocą klawiszy  ) może być poprawiona następująco:

- przesunąć kursor wprowadzania w błędne miejsce linii korzystając z klawiszy  
- usunąć błędne znaki klawiszem `DELETE+`,
- wstawić nowe znaki w miejsce wskazane kursorem,
- zakończyć poprawianie linii klawiszem `ENTER`.

#### postać zapisu pliku

W pamięci kasetowej mogą być zapisywane pliki następujących typów (rozdzielane atrybutem przy składowaniu poleceniem `SAVE`):

- pusty — treść programu w postaci znakowej,
- `CODE adr, adr` — postać binarna wskazanego adresami obszaru pamięci operacyjnej,
- `SCREEN$` — postać binarna obrazu aktualnie wyświetlanego na ekranie,
- `DATA nazwa__tabl()` — zawartość tablicy liczbowej lub tekstowej.

*słowo kluczowe*

Wszystkie słowa kluczowe języka są wprowadzane jako pojedyncze kody znaków — stąd są one rozróżnialne od każdego identyfikatora wprowadzanego znak po znaku.

*szer\_\_\_strefy*

Standardowa szerokość strefy na wydruku i ekranie wynosi 10 znaków.

*typ zmiennej i funkcji*

Wyróżnione są dwa typy zmiennych i funkcji:

- liczbowy (bez znacznika),
- \$ — tekstowy.

*wartość liczbowa*

Dopuszczalne są wartości liczbowe typu rzeczywistego z zakresu:

$$-10^{38} \dots -4 \cdot 10^{-39}, 0,4 \cdot 10^{-39} \dots 10^{38}$$

z dokładnością do 9 cyfr znaczących w postaci liczby całkowitej, ułamkowej lub wykładniczej z wykładnikiem identyfikowanym literą e lub E.

*wartość logiczna*

Wartość logiczna prawda (1), fałsz (0) jest uzyskiwana jako wynik wykorzystania →operatorów relacji i logicznych.

*wartość tekstowa*

Teksty mogą być dowolnej długości lub ograniczone deklaracją tablicy tekstowej, w której ostatni wymiar określa maksymalną długość tekstu. Znaki cudzysłowu mogą być umieszczone w tekście poprzez ich powtórzenie. Tekstowa zmienna wskaźnikowa bez wskaźnika oznacza zmienną tekstową o tym samym identyfikatorze.

*wywołanie systemu Basic*

System Basic zgłasza się po każdorazowym włączeniu komputera lub po wprowadzeniu polecenia NEW — wyprowadzając w ostatnim wierszu ekranu napis

© 1982 Sinclair Research Ltd.

W ZX Spectrum Plus istnieje klawisz RESET, umieszczony z lewej strony pod klawiaturą, zastępujący chwilowe wyłączenie-włączenie mikrokomputera.

*zaokrąglenie*

Wartość rzeczywista jest zaokrąglona do najbliższej wartości całkowitej nie większej niż dana wartość rzeczywista.



**Wykaz poleceń***Edycja programu**nr\_\_linii*

'usunięcie linii

LIST [*nr\_\_linii*]LLIST [*nr\_\_linii*]*Komentarz*REM *dowolny\_\_tekst**Uruchomienie programu*RUN [*nr\_\_linii*]

CONTINUE

*Ładowanie, składowanie programu*SAVE *spec\_\_pliku* [LINE *nr\_\_linii*]LOAD *spec\_\_pliku*VERIFY *spec\_\_pliku*MERGE *spec\_\_pliku**Definicje i deklaracje*DIM *nazwa\_\_tabl* (*rozm*[,*rozm*]...)DEF FN*nazwa\_\_fun* ([*par*[,*par*]...]) = *wyrażenie**Obliczenia arytmetyczne*FN*nazwa\_\_fun\_\_aryt* ([*arg*[,*arg*]...])ABS (*wyr\_\_aryt*)ACS (*wyr\_\_aryt*)ASN (*wyr\_\_aryt*)ATN (*wyr\_\_aryt*)COS (*wyr\_\_aryt*)EXP (*wyr\_\_aryt*)INT (*wyr\_\_aryt*)LN (*wyr\_\_aryt*)

PI

SGN (*wyr\_\_aryt*)SIN (*wyr\_\_aryt*)SQR (*wyr\_\_aryt*)TAN (*wyr\_\_aryt*)VAL (*wyr\_\_tekst*)

RND

LET *zm\_\_aryt* = *wyr\_\_aryt*RANDOMIZE [*wyr\_\_aryt*]

*Przetwarzanie tekstów*

FN*nazwa run\_tekst* ([*arg* [,*arg*]...])  
 CHR\$ (*kod\_znak*)  
 CODE (*wyr\_tekst*)  
 LEN (*wyr\_tekst*)  
 STR\$ (*wyr\_aryt*)  
 VAL\$ (*wyr\_tekst*)  
 LET *zm\_tekst* = *wyr\_tekst*

*Sterowanie programem*

STOP  
 GOTO *nr\_linii*  
 IF *wyr\_log* THEN *instrukcje*  
 FOR *zm\_ster* = *wart\_pocz* TO *wart\_konc* [STEP *wart\_kroku*]  
 NEXT *zm\_ster*  
 GOSUB *nr\_linii*  
 RETURN

*Wprowadzanie danych*

INKEY\$  
 DATA *stala* [,*stala*]...  
 INPUT [[*el\_wypr sep*]... *zmienna* [,*zmienna*]...]  
 INPUT LINE [*el\_wypr sep*]...*zm\_tekst*  
 RESTORE [*nr\_linii*]  
 READ *zmienna* [,*zmienna*]...

*Wyprowadzanie wyników*

ATTR (*nr\_wiersza*,*nr\_kol*)  
 SCREEN\$ (*nr\_wiersza*,*nr\_kol*)  
 CLS  
 PAUSE *okres*  
 INK *nr\_atramentu*  
 PAPER *nr\_koloru\_tla*  
 BORDER *nr\_koloru\_ramki*  
 FLASH *par018*  
 BRIGHT *par018*  
 OVER *par01*  
 INVERSE *par01*  
 PRINT [*el\_wypr sep*]...[*el\_wypr*]

```
LPRINT [el__wypr sep]...[el__wypr]
el__wypr < — AT nr__wiersza,nr__kol
                INK nr__koloru
                PAPER nr__koloru
                OVER par01
                FLASH par018
                BRIGHT par018
                INVERSE par01
                wyr__aryt
                wyr__tekst
                TAB nr__kol
```

### Grafika

```
■ POINT (wsp__x,wsp__y)
PLOT [OVER par01;] [INVERSE par01;] wsp__x,wsp__y
DRAW wsp__x,wsp__y[,kąt]
CIRCLE wsp__x,wsp__y, promień
COPY
SAVE spec__pliku SCREEN$
LOAD spec__pliku SCREEN$
```

### Współpraca z plikiem na kasecie

```
SAVE spec__pliku DATA nazwa__tab()
LOAD spec__pliku DATA nazwa__tab()
VERIFY spec__pliku DATA nazwa__tab()
```

### Łącze z zapisem wewnętrznym programu

```
■ PEEK (wyr__adr)
USR (wyr__adr)
USR (wyr__tekst)
■ INP (nr__portu)
CLEAR [wyr__adr]
OUT nr__portu,wyr__calk
SAVE spec__pliku CODE wyr__adr,wyr__adr
LOAD spec__pliku [CODE wyr__adr,wyr__adr]
VERIFY spec__pliku CODE wyr__adr,wyr__calk
```

### Generowanie dźwięku

```
BEEP okres,wyr__calk
```

**Wykaz komunikatów**

- 0 OK  
Poprawne zakończenie wykonania polecenia lub programu.
- 1 NEXT without FOR  
Brak ustalenia wartości początkowej zmiennej sterującej wskazanej przez instrukcję NEXT.
- 2 Variable not found  
Użycie zmiennej bez uprzedniego nadania jej wartości lub brak deklaracji tablicy przed użyciem zmiennej wskaźnikowej.
- 3 Subscript wrong  
Wartości wskaźników zmiennej wskaźnikowej spoza zakresu deklaracji rozmiarów tablicy, lub niezgodność liczby wskaźników i wymiarów tablicy.
- 4 Out of memory  
Przepełnienie pamięci.
- 5 Out of screen  
Próba wyświetlenia tekstu poza 22 wierszami ekranu lub wprowadzenia więcej niż 23 linii (w obu przypadkach określonych frazą AT).
- 6 Number too big  
Wynik obliczeń większy niż  $10^{38}$ .
- 7 RETURN without GO SUB  
Próba powrotu z podprogramu bez jego uprzedniego wywołania.
- 8 End of file  
Koniec pliku (jeżeli używamy Microdrive'a).
- 9 STOP statement  
Wstrzymanie wykonywania programu po wykonaniu instrukcji STOP — możliwość wznowienia poleceniem CONTINUE.
- A Invalid argument  
Błędny argument funkcji ASN, ACS, LN, SQR lub USR "...".
- B Integer out of range  
Niemożliwa konwersja wartości liczbowej rzeczywistej na wartość całkowitą ze względu na przekroczenie zakresu liczb całkowitych.
- C Nonsense in BASIC  
Argument funkcji VAL lub VAL\$ nie jest poprawny.
- D BREAK — CONT repeats  
Przerwano wykonywanie instrukcji wejścia-wyjścia — możliwość wznowienia poleceniem CONTINUE.

- E Out of DATA  
Brak danych dla instrukcji READ.
- F Invalid file name  
W zleceniu SAVE niepoprawna specyfikacja pliku.
- G No room for line  
Przepelnienie pamięci przy wprowadzaniu nowej linii programu.
- H STOP in INPUT  
We wprowadzonym tekście wykryto na początku zlecenie STOP — przerwanie wykonywania programu. Polecenie CONTINUE wznawia wykonywanie programu od powtórnego wykonania instrukcji INPUT.
- I FOR without NEXT  
Brak instrukcji kończącej cykl FOR, jeżeli nie może być ona ani razu wykonana.
- J Invalid I/O device  
Błąd w operacji wejścia-wyjścia.
- K Invalid colour  
Niepoprawny numer koloru.
- L BREAK into program  
Przerwano wykonywanie programu po instrukcji z linii o wskazanym numerze linii. Możliwość wznowienia wykonywania programu od następnej instrukcji zleceniem CONTINUE.
- M RAMTOP no good  
Zbyt duża lub zbyt mała wartość określająca adres ostatniego bajtu pola obszaru pamięci dla systemu BASIC.
- N Statement lost  
Powrót do nieistniejącej instrukcji.
- O Invalid stream  
Nieprawidłowy strumień dla operacji wejścia-wyjścia (przy obecności Microdrive'a).
- P FN without DEF  
Wywołanie niezdefiniowanej funkcji.
- Q Parameter error  
Nieprawidłowa liczba argumentów lub niezgodność typu przy wywołaniu funkcji.
- R Tape loading error  
Wykryto błąd podczas wczytywania lub weryfikacji treści programu lub danych zapisanych na taśmie kasetowej.

## 6. Wykaz słów kluczowych języka

ABS	ABSolutna wartość
AFTER	PO upływie czasu
AND	operator iloczynu logicznego I
ASC	kod znaku ASCII
ASN	Arc SiNus
AT	W wierszu, kolumnie
ATN	Arc TaNgens
ATTR	ATRYbuty znaku na ekranie
AUTO	AUTOMatyczna numeracja
BEEP	PIIP, ton
BIN	operator wartości BINarnej
BIN\$	BINarnie
BLOAD	ŁADUJ Binarny
BORDER	RAMKA ekranu
BRIGHT	JAŚNIEJszy znak
BSAVE	SKŁADUJ Binarny
CALL	WYWOŁAJ podprogram w języku wewnętrznym
CAT	KATalog wyświetl
CDBL	Konwersja do PODWójnej precyzji
CHAIN	ŁAŃCUCH programów
CHDIR	ZMiana KATalogu
CHR\$	ZNaK w kodzie ASCII
CINT	Konwersja do CAŁKowitej
CIRCLE	OKRĄG rysuj
CLEAR	CZYŚĆ, ZERUJ
CLG	CZyść ekran Graficzny
CLOSE	ZAMKNIJ plik
CLOSEIN	ZAMKNIJ WEjściowy
CLOSEOUT	ZAMKNIJ WYJściowy

CLR	CZYść
CLS	CZYść Ekran
CMD	POLeceNie przełączenia strumienia
CODE	KOD znaku ASCII
COLOR	KOLOR określ
COLOR =	KOLOR określ
COM	KOMunikacja
COMMON	WSPÓLNY obszar
CON	KONTynuuj
CONT	KONTynuuj
CONTINUE	KONTYNUUJ
COPY	KOPIUJ ekran
COS	COSinus
CREAL	Konwersja do RZECZywistej
CSNG	Konwersja do POJedynczej precyzji
CSRLIN	Aktualny WIERSZ Ekranu
CVD	Konwersja Zmiennej do poDwójnej precyzji
CVI	Konwersja Zmiennej do Całkowitej
CVS	Konwersja Zmiennej do poJedynczej precyzji
DATA	DANE
DATE\$	DATA tekstowo
DEC\$	DZIEŚiętnie
DEFDBL	DEFinicja poDWójNych precyzji
DEFFN	DEFinicja FuNkcji
DEFINT	DEFinicja CAŁKowitych
DEFREAL	DEFinicja RZECZywistych
DEFSNG	DEFinicja poJeDyNczych precyzji
DEFSTR	DEFinicja TEKSTowych
DEG	SToPnie (fun. tryg.)
DEL	USUń linie
DELETE	USUŃ linie
DI	Blokuj Przerwania
DIM	ROZmiar tablicy
DRAW	KREŚL linie, rysunek
DRAWR	KREŚL Względnie linie
DSP	PoKaZuj wartość zmiennej
EDIT	POPRAW linie
EI	Odblokuj Przerwania
ELSE	W PRZECiwnym razie
END	KONIEC
ENT	MeLodii Ton
ENU	MeLodii Głośność

ENVIRON	OTOCZENie programu modyfikuj
ENVIRON\$	OTOCZENia parametry
EOF	Koniec Pliku
EQV	operator równoważności
ERASE	WYMAŹ
ERL	numer Linii z BŁędem
ERR	Numer BŁędu
ERROR	BŁĄD
EVERY	po KAŻDYm upływie okresu czasu
EXP	EXPonent (exp(x))
FIELD	POLE bufora dla pliku
FILES	PLIKI
FIX	STAŁA część liczby
FLASH	MRUGAJ znakiem
FN_____	FUNKcji_____wywołanie
FOR	wyliczanie DLA
FRE	WOLNY obszar pamięci
GET	POBIERZ
GET#	POBIERZ ze strumienia
GOSUB	IDŹ do PODPRogramu
GO TO	IDŹ DO linii numer ...
GOTO	IDŹ DO linii numer ...
GR	GRafika na ekranie
HCOLOR =	KOLOR dla Wysokiej rozdzielczości
HEX\$	HEKSadecymalnie
HGR	GRafika o Wysokiej rozdzielczości
HGR2	(podobne do HGR)
HIMEM	GóRna granica PAMięci
HIMEM:	GóRna granica PAMięci
HLIN	RYSUJ linię Poziomo
HOME	POCZĄTEK ekranu (lewy, górny róg)
HPlot	RYSUJ kropkę lub linię dla Wysokiej rozd.
HTAB	TABulacja Pozioma
IF	JEŻELI
IMP	operator IMplikacji
IN	WProwadź z wejścia
IN#	wybór WEJścia (Ap)
INK	ATRAMENTu kolor
INKEY	WProwadź stan KLAWisza
INKEY\$	WProwadź kod KLAWisza
INP	WProwadź z wejścia
INPUT	WPROWADŹ



INPUT#	WPROWADŹ ze strumienia
INPUT\$	WPROWADŹ tekst
INSTR	WYSZUKAJ w TEKście fragment
INT	CAŁKOWITA część liczby
INVERSE	NEGATYW znaku, punktu
JOY	MANipulator (joystick)
KEY	KLAWISZ funkcyjny
KEY\$	KLAWISZ tekstowo
KILL	ZLIKWIDUJ plik
LEFT\$	LEWY fragment tekstu
LEN	DŁUGOŚĆ tekstu
LET	PRZYPISZ wartość
LINE	WIERSZ do wprowadzenia
LINE	LINIA do narysowania
LIST	LISTuj tekst programu
LLIST	DRUKUJ tekst programu
LN	Logarytm Naturalny
LOAD	ŁADUJ
LOC	LOKalizuj w pliku
LOCATE	UŁOKUJ kursor na ekranie
LOF	Długość Pliku
LOG	LOGarytm naturalny
LOG10	LOGarytm dziesiętny
LOMEM:	DOlna granica PAMIęci
LOWERS\$	MAŁE litery w tekście
LPOS	POZycja Drukowania
LPRINT	DRUKUJ
LSET	UMIEŚĆ Lewostronnie
MAN	RĘCZna numeracja
MAX	MAKSymalna z wartości
MEMORY	PAMIĘĆ — parametry
MERGE	ZŁÓŻ z dwóch
MID\$	ŚRODKowy fragment tekstu
MIN	MINimalna z wartości
MKDIR	UTwórz KATalog
MKDS	UtWórz z poDwójnej precyzji tekst
MKIS	UtWórz z Całkowitej tekst
MKSS	UtWórz z poJedynczej precyzji tekst
MOD	operator MODulo
MODE	TRYB funkcjonowania ekranu
MOVE	PRZESUŃ kursor graficzny
MOVER	PRZESUŃ Względnie kursor graficzny

NAME	NAZWIJ plik
NEXT	NASTĘPNY
NEW	NOWY tekst programu
NODSP	NIE PoKaZuj wartości
NORMAL	NORMALNIE (nie mrugaj, jaśniej..)
NOT	operator NEGacji
NOTRACE	BEZ ŚLEDZENIA
OCTS	OKTalnie
OFF	WYŁĄCZ działanie
ON	WŁĄCZ działanie
ON	GDY...
ON BREAK	GDY PRZERWANIE
ON ERROR	GDY BŁĄD
ON SQ	GDY Kolejka Dźwięków
ONERR	GDY BŁĄD
OPEN	OTWÓRZ plik
OPENIN	OTWÓRZ plik WEjściowy
OPENOUT	OTWÓRZ plik WYJściowy
OPTION	OPCJA
OR	operator SUMY logicznej
ORIGIN	POCZĄTEK współrzędnych
OUT	WYPRowadź na wyjście
OVER	NADRUK na ekranie
PRINT	MALUJ rysunek
PAPER	PAPIERu kolor
PAUSE	PAUZA
PDL	MANipulator
PEEK	pobranie zawartości bajtu z pamięci
PEN	PIÓRO — kolor
PI	3.14...
PLAY	GRAJ
PLOT	ZAZNACZ punkt
PLOTR	ZAZNACZ Względnie
POINT	PUNKTu atrybuty
POKE	wstawienie nowej zawartości do pamięci
POP	ZDEJMIJ ze stosu
POS	POZycja w wierszu
PR#	wybór WYJścia
PRESET	KASUJ punkt
PRINT	WYŚWIETL
PRINT#	WYPROWADŹ do strumienia
PSET	ZAZNACZ Punkt

PUT	WSTAW
RAD	RADiany (fun. tryg.)
RANDOMIZE	RANDOMIZACJA generatora
READ	CZYTAJ dane
RECALL	ODTWÓRZ z taśmy
RELEASE	USUŃ kanał dźwiękowy
REM	KOMentarz
REMAIN	POZOSTAŁY okres czasu
RENUM	PRZENUMeruj
RESET	ODŁĄCZ wszystkie pliki
RESTORE	ODTWÓRZ początek danych
RESUME	WRÓĆ z obsługi błędu
RETURN	POWRÓT z podprogramu
RIGHT\$	PRAWY fragment tekstu
RMDIR	USUŃ KATalog
RND	LOSowa liczba
ROT=	ROTacja rysunku
ROUND	ZAOKRąglij
RSET	UMIEŚĆ Prawostronnie
RUN	URUCHOM program
SAVE	SKŁADUJ program
SCALE=	SKALA rysunku
SCRN(	kolor punktu EKRaNu
SCREEN	tryb funkcjonowania EKRANu
SCREEN\$	EKRANu parametr
SGN	ZNaK liczby
SEG	SEGment pamięci
SHELL	uruchom pod swoją POWŁOKĄ drugi program
SHLOAD	ŁADUJ opis KSZtałtu
SIN	SINus
SOUND	DŹWIĘK — opis, wywołanie
SPACE\$	SPACJE
SPC	SPaCje
SPC(	SPaCje
SPEED	SZYBKość
SPEED=	SZYBKość wyprow.
SQ	Kolejka Dźwięków
SQR	PIERWiastek
STEP	KROK w instrukcji cyklu
STEP	WZGLĘDEM poprzedniego punktu
STICK	MANIPulator
STOP	STOP programu

STORE	ZAPAMIĘTAJ na taśmie
STR\$	TeKST
STRIG	Stan SPUSTu manipulatora
STRING\$	TEKST
SWAP	ZAMIEŃ
SYMBOL	SYMBOLu (znaku) definicja
SYS	powrót do SYSTEMu
SYSTEM	powrót do SYSTEMu
TAG	ZNACZNIK miejsca tekstu na ekranie
TAGOFF	ZNACZNIK WYŁĄCZ
TAB	TABulacja
TAB(	TABulacja
TAN	TANgens
TEST	TEST koloru atramentu
TSTR	TEST koloru w punkcie Względny
TEXT	tryb TEKSTowy ekranu
THEN	WTEDY
TIME	CZAS w jednostkach
TIMES	CZAS tekstowo
TIMER	STOPER
TO	DO podanej granicy
TRACE	ŚLEDZENIE włącz
TROFF	ŚLedzenie WYŁĄCZ
TRON	ŚLedzenie WŁĄCZ
UNT	konwersja do BEZznakowej Całkowitej
UPPER\$	DUŻE litery w tekście
USING	UŻYWAJĄC wzorca
USR	podprogram UŻyTkowy
VAL	WARTość liczbowa tekstu
VAL\$	WARTość tekstowa tekstu
VLIN	rysuj LINię Pionowo
VTAB	TABulacja Pionowa
VARPTR	ADRes ZMIennej
VARPTR\$	ADRes ZMIennej tekstowo
VERIFY	WERYFIKUJ
VIEW	OBRAZ ekranu
VPOS	POZycja Pionowa
WAIT	OCZEKUJ na we-wy
WEND	KONIEC cyklu Dopóki
WHILE	DOPÓKI
WIDTH	SZEROKOŚĆ wiersza
WINDOW	OKNO na ekranie

WRITE	ZAPISZ
WRITE#	ZAPISZ do strumienia
XDRAW	RYSUJ kształt
XPOS	POZycja współrzędnej X
XOR	operator SUMY modulo 2
YPOS	POZycja współrzędnej Y
ZONE	STREFA — szerokość

# Literatura

1. Amstrad BASIC: The Complete Technical Specification, AMSOFT.
2. Czech Z., Nałęcki K., Wołek B.: *Programowanie w języku BASIC*. wyd. 2, Warszawa, WNT 1985.
3. Goldstein L., Goldstein M.: IBM PC. An Introduction to the Operating System. BASIC Programming and Applications. New Jersey, Prentice-Hall.
4. Iszkowski W., Maniecki M.: *Programowanie w języku BASIC*. Warszawa, Wyd. 3, PWN 1986.
5. Iszkowski W.: *Nauka programowania w języku BASIC dla początkujących*. Warszawa, WNT 1987.
6. Poole L., McNiff M., Cook S.: *Apple II User's Guide*. Osborne-McGraw-Hill.
7. ZX Spectrum. Dokumentacja firmowa.

# *Mikrokomputery*

*Książki serii „Mikrokomputery” mają na celu prezentację i systematyzację wiedzy z zakresu systemów mikrokomputerowych (mikroprocesorowych). Na ich podstawie będzie możliwe: projektowanie, konstruowanie, modyfikowanie, naprawianie, rozszerzanie i adaptowanie do określonych zastosowań systemów mikrokomputerowych.*

*Problematyka serii obejmuje: elementy dyskretne i układy scalone cyfrowe stosowane w systemach mikrokomputerowych, metodologię projektowania i podstawowe algorytmy syntezy układów cyfrowych; rodziny mikroprocesorowe od strony architektury, jak i oprogramowania; układy sprzęgające mikrokomputery z urządzeniami zewnętrznymi, wskazówki praktyczne do projektowania sprzętu mikrokomputerowego, narzędzia i metodologię projektowania oprogramowania oraz zastosowanie systemów mikrokomputerowych.*

*Książki serii są przeznaczone dla szerokiego kręgu Czytelników — od niespecjalistów „wchodzących” w dziedzinę systemów mikrokomputerowych (informatycznych) do specjalistów poszukujących szczegółowych informacji o wskazanych problemach.*

# *Mikrokomputery*

W serii ukażą się m.in. następujące pozycje:

*J. Bielecki* — Fortran 77

*J. Bielecki* — Język C — interpretacja standardu

*J. Bielecki* — Język Forth

*J. Bielecki* — System operacyjny ISIS-II

*J. Bielecki* — Turbo Pascal wersja 3.0

*J. Bielecki* — Turbo Pascal z grafiką dla IBM PC

*J. Bielecki* — Wprowadzenie do języka C

*J. Boisgontier, S. Brebion* — Basic dla wszystkich

*W. Iszkowski* — Nauka programowania w języku BASIC dla początkujących

*J. Karczmarczuk* — Mikroprocesor Z80

*W. Link* — Jak mierzyć, sterować i regulować za pomocą Basicu?



**Wydawnictwa Naukowo-Techniczne planują wydanie następujących pozycji z dziedziny elektroniki:**

- B. Bany, W. Tęsny* — Diody i tyrystory mocy, wyd. 2
- W. Borodziejewicz, K. Jaszczak* — Cyfrowe przetwarzanie sygnałów. Wybrane zagadnienia
- J. W. Coffron, W. E. Long* — Technika sprzęgania układów w systemach mikroprocesorowych
- R. Ćwirko, M. Rusek, W. Marciniak* — Układy scalone w pytaniach i odpowiedziach, wyd. 2
- Z. Gutowski, M. Molski* — Komputer w pytaniach i odpowiedziach
- R. Hedtke* — Systemy mikroprocesorowe. Niezawodność, testowanie, tolerancja błędów
- Z. Korzec, Z. Ciota* — Układy z pojemnościami przełączanymi
- M. Morris Mano* — Architektura komputerów, wyd. 2
- H. Orłowski* — Komputerowe układy automatyki, wyd. 2
- S. Rosłonec* — Metody matematyczne w projektowaniu układów elektronicznych o parametrach rozłożonych
- U. Tietze, Ch. Schenk* — Układy półprzewodnikowe, wyd. 2
- R. Zielonko, A. Królikowski* — Metody pomiarowo-diagnostyczne analogowych układów elektronicznych
- Z. Żyszkowski* — Miernictwo akustyczne

WNT Warszawa 1987 r.  
Wydanie 1, Nakład 60.000 + 300 egz.  
Ark. wyd. 6,4. Ark. druk. 6,0  
Format B5. Papier offset, kl. III, 70 gr.  
Pracę oddano do składania w listopadzie 1986  
Podpisano do druku w czerwcu 1987. Druk ukończono w lipcu 1987  
Symbol Et 82180/WNT  
Drukarnia im. Rew. Październikowej Warszawa  
Zam. 3092/11/86. K-13



CENA ZŁ 300,-

Bonifikata 50%

250



ISBN 83-204-0809-1



# AMSTRAD

# CPC



**MÉMOIRE ÉCRITE**  
**MEMORY ENGRAVED**  
**MEMORIA ESCRITA**



<https://acpc.me/>

[FRA] Ce document a été préservé numériquement à des fins éducatives et d'études, et non commerciales.

[ENG] This document has been digitally preserved for educational and study purposes, not for commercial purposes.

[ESP] Este documento se ha conservado digitalmente con fines educativos y de estudio, no con fines comerciales.