

9/4.4.9

RSX opérations sur les nombres complexes

I. Opérations de base

Vous trouverez ici quatre RSX qui réalisent les opérations de base sur les nombres complexes : addition, soustraction, multiplication et division. Le lecteur relira avec profit le paragraphe 5/10.4.3 sur les espaces inconnus avant d'aborder ce chapitre.

RAPPELS

Les nombres complexes sont des entités mathématiques de la forme :

$$a + ib$$

où a et b sont des nombres réels, et i est un élément tel que $i \cdot i = -1$. a est appelé partie réelle du complexe, et b partie imaginaire.

Pour additionner deux nombres complexes, il suffit d'additionner séparément les parties réelle et imaginaire :

$$(a+ib) + (c+id) = (a+c) + i(b+d)$$

De même, pour soustraire deux nombres complexes, il suffit de soustraire séparément les parties réelle et imaginaire :

$$(a+ib) - (c+id) = (a-c) - i(b-d)$$

La multiplication de deux nombres complexes est moins immédiate. Elle résulte du développement des facteurs des deux nombres complexes :

$$(a+ib) * (c+id) = ac + iad + ibc + i^2bd = ac + iad + ibc - bd \text{ car } i^2 = -1 \\ (ac - bd) + i(ad + bc)$$

Enfin, la division de deux nombres complexes se calcule comme suit :

$$(a+ib) / (c+id) = (ac+bd) / (c^2 + d^2) + i(-ad+bc) / (c^2 + d^2)$$

Cette formule s'obtient par l'artifice suivant :

$$\begin{aligned} \frac{a + ib}{c + id} &= \frac{(a + ib)(c - id)}{(c + id)(c - id)} = \frac{(a + ib)(c - id)}{c^2 + d^2} \\ &= \frac{ac + ibc - aid + bd}{c^2 + d^2} \\ &= \frac{(ac + bd)}{c^2 + d^2} + i \frac{(bc - ad)}{c^2 + d^2} \end{aligned}$$

COMMENT UTILISER LE PROGRAMME

Les opérations ADD, SOU, MUL et DIV sont des RSX. Le programme qui gère leur fonctionnement est donc écrit en assembleur.

Voici son listing :

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; RSX ADD, SOUS, MUL et DIV
5          ; Format : !ADD,r1,i1,r2,i2
6          ;          !SOU,r1,i1,r2,i2
7          ;          !MUL,r1,i1,r2,i2
8          ;          !DIV,r1,i1,r2,i2
9          ; Entree : r1=partie reelle 1er nb
10         ;          i1=partie imagin 1er nb
11         ;          r2=partie reelle 2eme nb
12         ;          i2=partie imagin 2eme nb
13         ; Sortie : r1=partie reelle resultat
14         ;          i1=partie imagin resultat
15         ;-----
16         ;
17         ;
18         ;-----
19         ; Declaration des constantes
20         ; et des variables du programme
21         ;-----
22         ;
23         FLOENT: EQU 0BD67H          ;Flot->Entier

```

```

24          ADDFLO:    EQU  0BD79H          ;Addition flot
25          NEGFLO:    EQU  0BD8EH          ;Negation flot
26          MULFLO:    EQU  0BD82H          ;Multipl flot
27          DIVFLO:    EQU  0BD85H          ;Division flot
28          LOGEXT:    EQU  0BCD1H          ;KL LOG EXT
29          ;
30          BUF:       DS    4              ;ZONE RAM POUR LOG EXT
31 9004 1290  PTRTAB:   DW    TABLE        ;Pointeur TABLE
32 9006 C35590          JP    ADD           ;Calcul de l'addition
33 9009 C3A690          JP    SOU          ;Calcul de la soustractio
34 900C C30391          JP    MUL          ;Calcul de la multiplicat
35 900F C3BA91          JP    DIV          ;Calcul de la division
36 9012 4144  TABLE:  DB    "AD"
37 9014 C4             DB    "D"+80H
38 9015 534F          DB    "SO"
39 9017 D5             DB    "U"+80H
40 9018 4D55          DB    "MU"
41 901A CC             DB    "L"+80H
42 901B 4449          DB    "DI"
43 901D D6             DB    "V"+80H
44 901E 00             DB    0              ;Fin de table
45          ;
46          SAVRE:     DB    2              ;Adresse partie reelle
47          SAVIM:     DS    2              ;Adresse partie imagin
48          Z1:        DS    5              ;Zone reel 1
49          Z2:        DS    5              ;Zone reel 2
50          Z3:        DS    5              ;Zone reel 3
51          Z4:        DS    5              ;Zone reel 4
52          Z5:        DS    5              ;Zone reel 5
53          Z6:        DS    5              ;Zone reel 6
54          Z7:        DS    5              ;Zone reel 7
55          Z8:        DS    5              ;Zone reel 8
56          ;
57          ;-----
58          ; Definition des RSX
59          ;-----
60          ;

```

```

61          ;
62 903D DD6601      LD  H,(IX+1)
63 9040 DD6E00      LD  L,(IX+0)      ;Adresse de la var. y
64 9043 220090      LD  (SAVHL),HL      ;Sauvegarde
65 9046 CDDB90      CALL ZONE1      ;Memorisation
66 9049 DD6603      LD  H,(IX+3)
67 904C DD6E02      LD  L,(IX+2)      ;Adresse de la var. x
68 904F 220290      LD  (SAVHL2),HL      ;Sauvegarde
69 9052 CDE490      CALL ZONE2      ;Memorisation
70          ;
71 9055 210490      LD  HL,Z1
72 9058 110E90      LD  DE,Z3
73 905B CDED90      CALL FLODEHL      ;Sauvegarde
74 905E 210E90      LD  HL,Z3
75 9061 110E90      LD  DE,Z3
76 9064 CD82BD      CALL MULFLO      ; y^2
77 9067 210990      LD  HL,Z2
78 906A 111390      LD  DE,Z4
79 906D CDED90      CALL FLODEHL      ;Sauvegarde
80 9070 211390      LD  HL,Z4
81 9073 111390      LD  DE,Z4
82 9076 CD82BD      CALL MULFLO      ; x^2
83 9079 210E90      LD  HL,Z3
84 907C 111390      LD  DE,Z4
85 907F CD79BD      CALL ADDFLO      ; x^2+y^2
86 9082 210E90      LD  HL,Z3
87 9085 CD9ABD      CALL RACFLO      ; Rac(x^2+y^2)
88          ;
89 9088 210E90      LD  HL,Z3
90 908B ED5B0290     LD  DE,(SAVHL2)
91 908F CDED90      CALL FLODEHL      ;Module

```

```

92          ;
93          ;-----
94          ; Calcul de l'addition
95          ;-----
96          ;
97 907F 212390      LD   HL,Z1
98 9082 112D90      LD   DE,Z3
99 9085 CD79BD      CALL ADDFLO      ;Addition partie reelle
100 9088 212390     LD   HL,Z1
101 908B ED5B1F90   LD   DE,(SAVRE)
102 908F CD9292     CALL FLODEHL     ;Sauvegarde partie reelle
103 9092 212B90     LD   HL,Z2
104 9095 113290     LD   DE,Z4
105 9098 CD79BD     CALL ADDFLO     ;Addition partie imagin
106 909B 212B90     LD   HL,Z2
107 909E ED5B2190  LD   DE,(SAVIM)
108 90A2 CD9292     CALL FLODEHL     ;Sauvegarde partie imagin
109 90A5 C9         RET
110          ;
111          ;-----
112          ; Traitement de SOU
113          ;-----
114          ;
115          SOU:      EQU  $
116          ;
117          ;-----
118          ; Lecture des donnees passees
119          ;-----
120          ;

```

```

121 90A6 DD6601      LD   H, (IX+1)
122 90A9 DD6E00      LD   L, (IX+0)
123 90AC CD6592      CALL ZONE4          ;Memo 4eme argument
124 90AF DD6603      LD   H, (IX+3)
125 90B2 DD6E02      LD   L, (IX+2)
126 90B5 CD5C92      CALL ZONE3          ;Memo 3eme argument
127 90B8 DD6605      LD   H, (IX+5)
128 90BB DD6E04      LD   L, (IX+4)
129 90BE 222190      LD   (SAVIM),HL    ;Sauvegarde imagin
130 90C1 CD5392      CALL ZONE2          ;Memo 2eme argument
131 90C4 DD6607      LD   H, (IX+7)
132 90C7 DD6E06      LD   L, (IX+6)
133 90CA 221F90      LD   (SAVRE),HL    ;Sauvegarde reelle
134 90CD CD4A92      CALL ZONE1          ;Memo 1er argument
135                   ;
136                   ;-----
137                   ; Calcul de la soustraction
138                   ;-----
139                   ;
140 90D0 212D90      LD   HL,Z3
141 90D3 CD8EBD      CALL NEGFLD
142 90D6 212390      LD   HL,Z1
143 90D9 112D90      LD   DE,Z3
144 90DC CD79BD      CALL ADDFLD        ;Addition partie reelle
145 90DF 212390      LD   HL,Z1
146 90E2 EDSB1F90    LD   DE, (SAVRE)
147 90E6 CD9292      CALL FLODEHL       ;Sauvegarde partie reelle
148 90E9 213290      LD   HL,Z4
149 90EC CD8EBD      CALL NEGFLD
150 90EF 212890      LD   HL,Z2
151 90F2 113290      LD   DE,Z4

```

```

152 90F5 CD79BD          CALL ADDFLO           ;Addition partie imagin
153 90FB 212890          LD   HL,Z2
154 90FB ED5B2190        LD   DE,(SAVIM)
155 90FF CD9292          CALL FLODEHL         ;Sauvegarde partie imagin
156 9102 C9              RET

157                      ;
158                      ;
159                      ;-----
160                      ; Traitement de MUL
161                      ;-----
162                      ;
163 MUL:                  EQU $
164                      ;
165                      ;-----
166                      ; Lecture des donnees passees
167                      ;-----
168                      ;

169 9103 DD6601          LD   H,(IX+1)
170 9106 DD6E00          LD   L,(IX+0)
171 9109 CD6592          CALL ZONE4           ;Memo 4eme argument
172 910C DD6603          LD   H,(IX+3)
173 910F DD6E02          LD   L,(IX+2)
174 9112 CD5C92          CALL ZONE3           ;Memo 3eme argument
175 9115 DD6605          LD   H,(IX+5)
176 9118 DD6E04          LD   L,(IX+4)
177 911B 222190          LD   (SAVIM),HL      ;Sauvegarde imagin
178 911E CD5392          CALL ZONE2           ;Memo 2eme argument
179 9121 DD6607          LD   H,(IX+7)
180 9124 DD6E06          LD   L,(IX+6)

```

```

181 9127 221F90          LD  (SAVRE),HL          ;Sauvegarde reelle
182 912A CD4A92          CALL ZONE1              ;Memo 1er argument
183                      ;
184                      ;-----
185                      ; Calcul de la multiplication
186                      ;-----
187                      ;
188 912D 212390          LD  HL,Z1
189 9130 CD7792          CALL ZONE6              ;Copie de Z1 -> Z6
190 9133 213C90          LD  HL,Z6
191 9136 112D90          LD  DE,Z3
192 9139 CD82BD          CALL MULFLO            ;Z6<-Z1*Z3
193 913C 212890          LD  HL,Z2
194 913F CD8092          CALL ZONE7              ;Copie de Z2 -> Z7
195 9142 214190          LD  HL,Z7
196 9145 113290          LD  DE,Z4
197 9148 CD82BD          CALL MULFLO            ;Z7<-Z2*Z4
198 914B 214190          LD  HL,Z7
199 914E CD8E8D          CALL NEGFLD            ;Z7<--Z2*Z4
200 9151 213C90          LD  HL,Z6
201 9154 114190          LD  DE,Z7
202 9157 CD79BD          CALL ADDFLO            ; Z6<-Z1*Z3-Z2*Z4
203 915A 213C90          LD  HL,Z6
204 915D ED5B1F90        LD  DE,(SAVRE)
205 9161 CD9292          CALL FLODEHL           ;Sauvegarde partie reelle
206
207 9164 212390          LD  HL,Z1
208 9167 113290          LD  DE,Z4
209 916A CD82BD          CALL MULFLO            ;Z1<-Z1*Z4
210 916D 212890          LD  HL,Z2
211 9170 112D90          LD  DE,Z3

```



```

212 9173 CD62BD          CALL MULFLO          ; Z2<-Z2*Z3
213 9176 212390         LD   HL,Z1
214 9179 112890         LD   DE,Z2
215 917C CD79BD          CALL ADDFLO          ; Z1<-Z1*Z4-Z2*Z3
216 917F 212390         LD   HL,Z1
217 9182 ED5B2190       LD   DE,(SAVIM)
218 9186 CD9292         CALL FLODEHL        ;Sauvegarde partie imagin
219 9189 C9             RET
220                    ;
221                    ;-----
222                    ; Traitement de DIV
223                    ;-----
224                    ;
225                    DIV:      EQU  $
226                    ;
227                    ;-----
228                    ; Lecture des donnees passees
229                    ;-----
230                    ;
231 918A DD6601          LD   H,(IX+1)
232 918D DD6E00         LD   L,(IX+0)
233 9190 CD6592         CALL ZONE4          ;Memo 4eme argument
234 9193 DD6603         LD   H,(IX+3)
235 9196 DD6E02         LD   L,(IX+2)
236 9199 CD5C92         CALL ZONE3          ;Memo 3eme argument
237 919C DD6605         LD   H,(IX+5)
238 919F DD6E04         LD   L,(IX+4)
239 91A2 222190         LD   (SAVIM),HL    ;Sauvegarde imagin
240 91A5 CD5392         CALL ZONE2          ;Memo 2eme argument

```

```

241 91A8 DD6607          LD   H,(IX+7)
242 91AB DD6E06          LD   L,(IX+6)
243 91AE 221F90          LD   (SAVRE),HL          ;Sauvegarde reelle
244 91B1 CD4A92          CALL ZONE1              ;Memo ier argument
245                      ;
246                      ;-----
247                      ; Calcul de la division
248                      ;-----
249                      ;
250 91B4 212390          LD   HL,Z1
251 91B7 CD7792          CALL ZONE6              ;Copie de Z1 -> Z6
252 91BA 213C90          LD   HL,Z6
253 91BD 112D90          LD   DE,Z3
254 91C0 CD82BD          CALL MULFLO            ;Z6<-Z1*Z3
255 91C3 212890          LD   HL,Z2
256 91C6 CDB092          CALL ZONE7              ;Copie de Z2 -> Z7
257 91C9 214190          LD   HL,Z7
258 91CC 113290          LD   DE,Z4
259 91CF CD82BD          CALL MULFLO            ;Z7<-Z2*Z4
260 91D2 213C90          LD   HL,Z6
261 91D5 114190          LD   DE,Z7
262 91D8 CD79BD          CALL ADDFLO            ; Z6<-Z1*Z3+Z2*Z4
263 91DB 212D90          LD   HL,Z3
264 91DE CDB092          CALL ZONE7
265 91E1 214190          LD   HL,Z7
266 91E4 114190          LD   DE,Z7
267 91E7 CD82BD          CALL MULFLO            ;Z7<-Z3*Z3
268 91EA 213290          LD   HL,Z4
269 91ED CDB992          CALL ZONE8
270 91F0 214690          LD   HL,Z8
271 91F3 114690          LD   DE,Z8

```

```

272 91F6 CD82BD      CALL MULFLO          ;Z8<-Z4*Z4
273 91F9 214190      LD   HL,Z7
274 91FC 114690      LD   DE,Z8
275 91FF CD79BD      CALL ADDFLO          ;Z7<-Z3*Z3+Z4*Z4
276 9202 213C90      LD   HL,Z6
277 9205 114190      LD   DE,Z7
278 9208 CD85BD      CALL DIVFLO          ;Z6<-Partie reelle repons
279 920B 213C90      LD   HL,Z6
280 920E ED5B1F90     LD   DE,(SAVRE)
281 9212 CD9292      CALL FLODEHL         ;Sauvegarde partie reelle
282
283 9215 212390      LD   HL,Z1
284 9218 113290      LD   DE,Z4
285 921B CD82BD      CALL MULFLO          ;Z1<-Z1*Z4
286 921E 212390      LD   HL,Z1
287 9221 CD8E8D      CALL NEGFLO          ;Z1<--Z1*Z4
288 9224 212890      LD   HL,Z2
289 9227 112D90      LD   DE,Z3
290 922A CD82BD      CALL MULFLO          ;Z2<-Z2*Z3
291 922D 212390      LD   HL,Z1
292 9230 112890      LD   DE,Z2
293 9233 CD79BD      CALL ADDFLO          ;Z1<--Z1*Z4+Z2*Z3
294 9236 212390      LD   HL,Z1
295 9239 114190      LD   DE,Z7
296 923C CD85BD      CALL DIVFLO          ;Z1<-Partie imagin repons
297 923F 212390      LD   HL,Z1
298 9242 ED5B2190     LD   DE,(SAVIM)
299 9246 CD9292      CALL FLODEHL         ;Sauvegarde partie imagin
300 9249 C9          RET

```

```
301          ;
302          ;-----
303          ; Zone des sous-programmes
304          ;-----
305          ;
306          ;-----
307          ; Transfert des BC octets pointes
308          ; par HL dans Z1, Z4 ou Z5
309          ;-----
310          ;
311          ZONE1:      EQU  $
312 924A 112390        LD  DE,Z1
313 924D 010500        LD  BC,5
314 9250 EDB0         LDIR
315 9252 C9           RET
316          ;
317          ZONE2:      EQU  $
318 9253 112890        LD  DE,Z2
319 9256 010500        LD  BC,5
320 9259 EDB0         LDIR
321 925B C9           RET
322          ;
323          ZONE3:      EQU  $
324 925C 112D90        LD  DE,Z3
325 925F 010500        LD  BC,5
326 9262 EDB0         LDIR
327 9264 C9           RET
328          ;
329          ZONE4:      EQU  $
330 9265 113290        LD  DE,Z4
331 9268 010500        LD  BC,5
```

```

332 926B EDB0.          LDIR
333 926D C9            RET
334                    ;
335                    ZONE5: EQU $
336 926E 113790        LD DE,Z5
337 9271 010500        LD BC,5
338 9274 EDB0          LDIR
339 9276 C9            RET
340                    ;
341                    ZONE6: EQU $
342 9277 113C90        LD DE,Z6
343 927A 010500        LD BC,5
344 927D EDB0          LDIR
345 927F C9            RET
346                    ;
347                    ZONE7: EQU $
348 9280 114190        LD DE,Z7
349 9283 010500        LD BC,5
350 9286 EDB0          LDIR
351 9288 C9            RET
352                    ;
353                    ZONE8: EQU $
354 9289 114690        LD DE,Z8
355 928C 010500        LD BC,5
356 928F EDB0          LDIR
357 9291 C9            RET
358                    ;
359                    ;-----
360                    ; Transfert flottant de (HL)

```

```

361          ; dans (DE)
362          ; -----
363          ;
364          FLODEHL: EQU $
365 9292 010500          LD  BC,5
366 9295 EDB0          LDIR
367 9297 C9           RET
368          END

```

Avant d'utiliser les RSX, il faut les initialiser à l'aide d'une instruction CALL :

CALL &904B

Les quatre RSX fonctionnent de la même manière.

Les parties réelle et imaginaire des deux nombres à additionner, soustraire, multiplier ou diviser sont placées dans des variables réelles et transmises à la RSX par leur adresse. Par exemple pour !MUL :

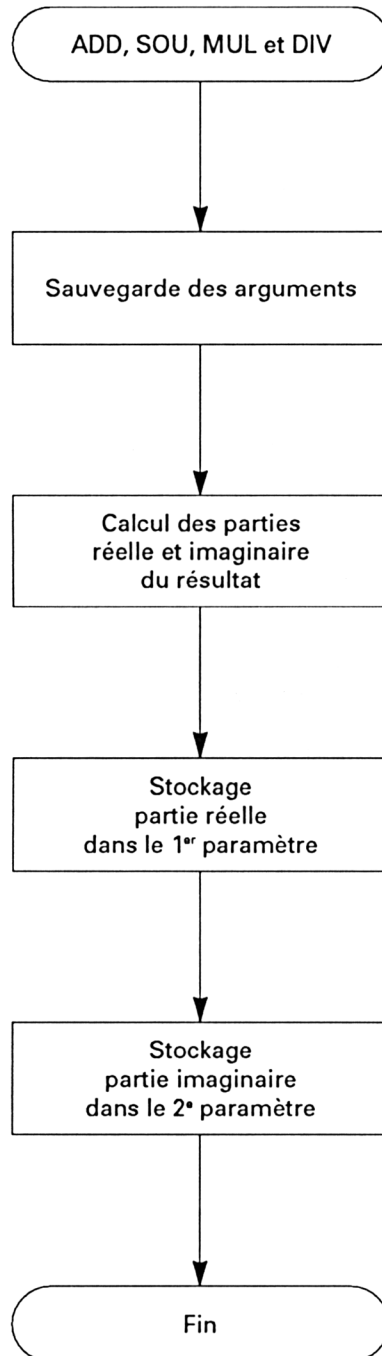
a=4.2 : b=-1 : c=10 : d=3.25
MUL, @a, @b; @c, @d

Le résultat est renvoyé dans les deux premières variables. Dans notre exemple :

- la variable a contient la partie réelle du résultat ;
- la variable b contient la partie imaginaire du résultat.

LES RSX EN DÉTAIL

La logique de fonctionnement des RSX apparaît dans l'ordinogramme suivant :



L'en-tête du programme précise le mode d'appel des RSX, les paramètres en entrée et en sortie.

Les RSX utilisent des routines de traitement de nombres réels dont les points d'entrée varient en fonction de l'ordinateur utilisé :

Point d'entrée	CPC 464	CPC 664	CPC 6128
FLOENT	0BD46H	0BD67H	0BD6AH
ADDFLO	0BD58H	0BD79H	0BD7CH
NEGFLO	0BD6DH	0BD8EH	0BD91H
MULFLO	0BD61H	0BD82H	0BD85H
DIVFLO	0BD64H	0BD85H	0BD88H

Remarques :

- Pour information, le programme a été écrit sur un CPC 664.
- Pour faciliter l'utilisation des RSX sur les trois types d'ordinateurs, les chargeurs Basic ont été écrits dans les versions 464, 664 et 6128.

Reportez-vous à la partie 9, chapitre 4.4.1 si vous désirez avoir des détails sur la mise en œuvre de RSX (macro LOGEXT, buffers BUF, PTRTAB et TABLE, routine DEFRSX).

Les zones Z1 à Z8 de 5 octets permettent de manipuler des nombres réels à travers les routines système FLOENT, ADDFLO, NEGFLO, MULFLO et DIVFLO et les sous-programmes ZONE1 à ZONE8 et FLODEHL.

Nous allons étudier en détails le traitement lié à la multiplication de deux nombres complexes. Les traitements des trois autres opérations sont très similaires.

Lorsque la RSX !MUL est identifiée par le Basic, la routine de traitement située à l'étiquette MUL est exécutée. La première action effectuée dans cette routine consiste à sauvegarder les paramètres passés à la RSX par l'intermédiaire de la pile :

```
LD      H, (IX+1)
LD      L, (IX+0)
CALL   ZONE4      ;Memo 4eme argument
...
LD      H, (IX+7)
LD      L, (IX+6)
LD      (SAVRE), HL ;Sauvegarde reelle
CALL   ZONE1      ;Memo 1er argument
```


Les procédures **ZONE1** à **ZONE4** mémorisent les paramètres 1 à 4 dans les buffers Z1 à Z4. L'adresse du premier paramètre est sauvegardée dans la variable **SAVRE**, et celle du second paramètre dans la variable **SAVIM**. Ces deux adresses permettront à la RSX de stocker la partie réelle du résultat dans le premier argument et la partie imaginaire du résultat dans le second argument.

Lorsque les arguments sont mémorisés, le programme effectue les opérations nécessaires en utilisant les routines réelles **MULFLO**, **ADDFO** et **NEGFLO** (lignes 188 à 218). La partie réelle du résultat est mémorisée dans le premier argument à l'aide du sous-programme **FLODEHL** (lignes 203 à 205) :

```
LD      HL, Z6
LD      DE, (SAVRE)
CALL    FLODEHL      ;Sauvegarde partie réelle
```

Des instructions du même type sauvegardent la partie imaginaire du résultat dans le second argument (lignes 216 à 218).

LES CHARGEURS BASIC

Si vous préférez utiliser un chargeur Basic, voici le listing correspondant, pour les trois versions de CPC, et les données de checksum afférentes.

Version CPC 464 :

```
1000 '=====
1010 ' Demonstration des RSX ADD, SOU, MUL et DIV
1020 '=====
1030 ' Version CPC 464
1040 '=====
1050 '
1060 FOR i=&9000 TO &9297
1070   READ a$
1080   a$="&"+a$
1090   a=VAL(a$)
1100   POKE i,a
1110 NEXT i
1120 '
1130 '-----
1140 ' Initialisation des RSX
1150 '-----
1160 '
1170 CALL &904B
1180 '
1190 '-----
```

```

1200 ' Demonstration
1210 '-----
1220 CLS
1230 PRINT"Utilisation des RSX !ADD, !SOU, !MUL et !DIV"
1240 PRINT
1250 A=10:B=5:C=-3:D=4.25
1260 PRINT "("A"+"B"i) + ("C"+"D"i) = ";
1270 !ADD,@A,@B,@C,@D
1280 PRINT A;
1290 IF B>=0 THEN PRINT"+";
1300 PRINT B"i"
1310 '
1320 A=10:B=5:C=-3:D=4.25
1330 PRINT "("A"+"B"i) - ("C"+"D"i) = ";
1340 !SOU,@A,@B,@C,@D
1350 PRINT A;
1360 IF B>=0 THEN PRINT"+";
1370 PRINT B"i"
1380 '
1390 A=10:B=5:C=-3:D=4.25
1400 PRINT "("A"+"B"i) * ("C"+"D"i) = ";
1410 !MUL,@A,@B,@C,@D
1420 PRINT A;
1430 IF B>=0 THEN PRINT"+";
1440 PRINT B"i"
1450 '
1460 A=10:B=5:C=-3:D=4.25
1470 PRINT "("A"+"B"i) / ("C"+"D"i) = ";
1480 !DIV,@A,@B,@C,@D
1490 PRINT A;
1500 IF B>=0 THEN PRINT"+";
1510 PRINT B"i"
1520 '
1530 END
1540 '
1550 '-----
1560 ' Donnees des RSX ADD, SOU, MUL et DIV
1570 '-----
1580 '
1590 DATA 0,0,0,0,12,90,C3,55,90,C3,A6,90,C3,3,91,C3
1600 DATA 8A,91,41,44,C4,53,4F,D5,4D,55,CC,44,49,D6,0,0
1610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1630 DATA 0,0,0,0,0,0,0,0,0,0,0,1,4,90,21,0
1640 DATA 90,CD,D1,BC,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66
1650 DATA 3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90
1660 DATA CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21
1670 DATA 23,90,11,2D,90,CD,58,BD,21,23,90,ED,5B,1F,90,CD
1680 DATA 92,92,21,28,90,11,32,90,CD,58,BD,21,28,90,ED,5B
1690 DATA 21,90,CD,92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD

```

```

1700 DATA 66,3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21
1710 DATA 90,CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92
1720 DATA 21,2D,90,CD,6D,8D,21,23,90,11,2D,90,CD,58,8D,21
1730 DATA 23,90,ED,5B,1F,90,CD,92,92,21,32,90,CD,6D,8D,21
1740 DATA 28,90,11,32,90,CD,58,8D,21,28,90,ED,5B,21,90,CD
1750 DATA 92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66,3,DD
1760 DATA 6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90,CD,53
1770 DATA 92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21,23,90
1780 DATA CD,77,92,21,3C,90,11,2D,90,CD,61,8D,21,28,90,CD
1790 DATA 80,92,21,41,90,11,32,90,CD,61,8D,21,41,90,CD,6D
1800 DATA 8D,21,3C,90,11,41,90,CD,58,8D,21,3C,90,ED,5B,1F
1810 DATA 90,CD,92,92,21,23,90,11,32,90,CD,61,8D,21,28,90
1820 DATA 11,2D,90,CD,61,8D,21,23,90,11,28,90,CD,58,8D,21
1830 DATA 23,90,ED,5B,21,90,CD,92,92,C9,DD,66,1,DD,6E,0
1840 DATA CD,65,92,DD,66,3,DD,6E,2,CD,5C,92,DD,66,5,DD
1850 DATA 6E,4,22,21,90,CD,53,92,DD,66,7,DD,6E,6,22,1F
1860 DATA 90,CD,4A,92,21,23,90,CD,77,92,21,3C,90,11,2D,90
1870 DATA CD,61,8D,21,28,90,CD,80,92,21,41,90,11,32,90,CD
1880 DATA 61,8D,21,3C,90,11,41,90,CD,58,8D,21,2D,90,CD,80
1890 DATA 92,21,41,90,11,41,90,CD,61,8D,21,32,90,CD,89,92
1900 DATA 21,46,90,11,46,90,CD,61,8D,21,41,90,11,46,90,CD
1910 DATA 58,8D,21,3C,90,11,41,90,CD,64,8D,21,3C,90,ED,5B
1920 DATA 1F,90,CD,92,92,21,23,90,11,32,90,CD,61,8D,21,23
1930 DATA 90,CD,6D,8D,21,28,90,11,2D,90,CD,61,8D,21,23,90
1940 DATA 11,28,90,CD,58,8D,21,23,90,11,41,90,CD,64,8D,21
1950 DATA 23,90,ED,5B,21,90,CD,92,92,C9,11,23,90,1,5,0
1960 DATA ED,B0,C9,11,28,90,1,5,0,ED,B0,C9,11,2D,90,1
1970 DATA 5,0,ED,B0,C9,11,32,90,1,5,0,ED,B0,C9,11,37
1980 DATA 90,1,5,0,ED,B0,C9,11,3C,90,1,5,0,ED,B0,C9
1990 DATA 11,41,90,1,5,0,ED,B0,C9,11,46,90,1,5,0,ED
2000 DATA B0,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0

```

Checksum :

```

63 B2 0 0 B6 52 7B EE 2 D9 A3 51 5E 80 9D 13 6B B9 81 29 F4 C8 F2 5F FC 3F D8 A4
3C 1 23 75 E 7C F3 76 36 70 F7 4B 2D E8

```

Version CPC 664 :

```

1000 '=====
1010 ' Demonstration des RSX ADD, SOU, MUL et DIV
1020 '=====
1030 ' Version CPC 664
1040 '=====
1050 '
1060 FOR i=&9000 TO &9297
1070   READ a$
1080   a$="&" + a$
1090   a=VAL(a$)
1100   POKE i,a
1110 NEXT i
1120 '
1130 '-----
1140 ' Initialisation des RSX
1150 '-----
1160 '
1170 CALL &904B
1180 '
1190 '-----
1200 ' Demonstration
1210 '-----
1220 CLS
1230 PRINT"Utilisation des RSX IADD, ISOU, IMUL et IDIV"
1240 PRINT
1250 A=10;B=5;C=-3;D=4.25
1260 PRINT "("A"+"B"i) + ("C"+"D"i) = ";
1270 IADD,@A,@B,@C,@D
1280 PRINT A;
1290 IF B>=0 THEN PRINT"+";
1300 PRINT B"i"
1310 '
1320 A=10;B=5;C=-3;D=4.25
1330 PRINT "("A"+"B"i) - ("C"+"D"i) = ";
1340 ISOU,@A,@B,@C,@D
1350 PRINT A;
1360 IF B>=0 THEN PRINT"+";
1370 PRINT B"i"
1380 '
1390 A=10;B=5;C=-3;D=4.25
1400 PRINT "("A"+"B"i) * ("C"+"D"i) = ";
1410 IMUL,@A,@B,@C,@D
1420 PRINT A;
1430 IF B>=0 THEN PRINT"+";
1440 PRINT B"i"
1450 '
1460 A=10;B=5;C=-3;D=4.25

```

```

1470 PRINT "("A"+"B"i) / ("C"+"D"i) = ";
1480 !DIV,@A,@B,@C,@D
1490 PRINT A;
1500 IF B>=0 THEN PRINT"+";
1510 PRINT B"i"
1520 '
1530 END
1540 '
1550 '-----
1560 ' Donnees des RSX ADD, SOU, MUL et DIV
1570 '-----
1580 '
1590 DATA 0,0,0,0,12,90,C3,55,90,C3,A6,90,C3,3,91,C3
1600 DATA 8A,91,41,44,C4,53,4F,D5,4D,55,CC,44,49,D6,0,0
1610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1630 DATA 0,0,0,0,0,0,0,0,0,0,0,1,4,90,21,0
1640 DATA 90,CD,D1,BC,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66
1650 DATA 3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90
1660 DATA CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21
1670 DATA 23,90,11,2D,90,CD,79,BD,21,23,90,ED,5B,1F,90,CD
1680 DATA 92,92,21,28,90,11,32,90,CD,79,BD,21,28,90,ED,5B
1690 DATA 21,90,CD,92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD
1700 DATA 66,3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21
1710 DATA 90,CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92
1720 DATA 21,2D,90,CD,6E,BD,21,23,90,11,2D,90,CD,79,BD,21
1730 DATA 23,90,ED,5B,1F,90,CD,92,92,21,32,90,CD,6E,BD,21
1740 DATA 28,90,11,32,90,CD,79,BD,21,28,90,ED,5B,21,90,CD
1750 DATA 92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66,3,DD
1760 DATA 6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90,CD,53
1770 DATA 92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21,23,90
1780 DATA CD,77,92,21,3C,90,11,2D,90,CD,82,BD,21,28,90,CD
1790 DATA 80,92,21,41,90,11,32,90,CD,82,BD,21,41,90,CD,8E
1800 DATA BD,21,3C,90,11,41,90,CD,79,BD,21,3C,90,ED,5B,1F
1810 DATA 90,CD,92,92,21,23,90,11,32,90,CD,82,BD,21,28,90
1820 DATA 11,2D,90,CD,82,BD,21,23,90,11,28,90,CD,79,BD,21
1830 DATA 23,90,ED,5B,21,90,CD,92,92,C9,DD,66,1,DD,6E,0
1840 DATA CD,65,92,DD,66,3,DD,6E,2,CD,5C,92,DD,66,5,DD
1850 DATA 6E,4,22,21,90,CD,53,92,DD,66,7,DD,6E,6,22,1F
1860 DATA 90,CD,4A,92,21,23,90,CD,77,92,21,3C,90,11,2D,90
1870 DATA CD,82,BD,21,28,90,CD,80,92,21,41,90,11,32,90,CD
1880 DATA 82,BD,21,3C,90,11,41,90,CD,79,BD,21,2D,90,CD,80
1890 DATA 92,21,41,90,11,41,90,CD,82,BD,21,32,90,CD,89,92
1900 DATA 21,46,90,11,46,90,CD,82,BD,21,41,90,11,46,90,CD
1910 DATA 79,BD,21,3C,90,11,41,90,CD,85,BD,21,3C,90,ED,5B
1920 DATA 1F,90,CD,92,92,21,23,90,11,32,90,CD,82,BD,21,23
1930 DATA 90,CD,8E,BD,21,28,90,11,2D,90,CD,82,BD,21,23,90

```

```

1940 DATA 11,28,90,CD,79,BD,21,23,90,11,41,90,CD,85,BD,21
1950 DATA 23,90,ED,5B,21,90,CD,92,92,C9,11,23,90,1,5,0
1960 DATA ED,B0,C9,11,28,90,1,5,0,ED,B0,C9,11,2D,90,1
1970 DATA 5,0,ED,B0,C9,11,32,90,1,5,0,ED,B0,C9,11,37
1980 DATA 90,1,5,0,ED,B0,C9,11,3C,90,1,5,0,ED,B0,C9
1990 DATA 11,41,90,1,5,0,ED,B0,C9,11,46,90,1,5,0,ED
2000 DATA B0,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0

```

Checksum :

```

63 B2 0 0 B6 52 7B EE 23 FA A3 51 5E C2 BE 34 6B BB 81 4A 37 E9 14 A1 FC 3F D9 A
4 5D 43 44 96 50 9D 36 B8 36 70 F7 4B 2D EB

```

Version CPC 6128 :

```

1000 '=====
1010 ' Demonstration des RSX ADD, SOU, MUL et DIV
1020 '=====
1030 ' Version CPC 6128
1040 '=====
1050 '
1060 FOR i=&9000 TO &9297
1070   READ a$
1080   a$="&"+a$
1090   a=VAL(a$)
1100   POKE i,a
1110 NEXT i
1120 '
1130 '-----
1140 ' Initialisation des RSX
1150 '-----
1160 '
1170 CALL &904B
1180 '
1190 '-----
1200 ' Demonstration
1210 '-----
1220 CLS
1230 PRINT"Utilisation des RSX !ADD, !SOU, !MUL et !DIV"
1240 PRINT
1250 A=10:B=5:C=-3:D=4.25
1260 PRINT "("A"+"B"i) + ("C"+"D"i) = ";
1270 !ADD,@A,@B,@C,@D
1280 PRINT A;
1290 IF B>=0 THEN PRINT"+";
1300 PRINT B"i"

```

```

1310 '
1320 A=10:B=5:C=-3:D=4.25
1330 PRINT "("A"+"B"i) - ("C"+"D"i) = ";
1340 !SCU,@A,@B,@C,@D
1350 PRINT A;
1360 IF B>=0 THEN PRINT"+";
1370 PRINT B"i"
1380 '
1390 A=10:B=5:C=-3:D=4.25
1400 PRINT "("A"+"B"i) * ("C"+"D"i) = ";
1410 !MUL,@A,@B,@C,@D
1420 PRINT A;
1430 IF B>=0 THEN PRINT"+";
1440 PRINT B"i"
1450 '
1460 A=10:B=5:C=-3:D=4.25
1470 PRINT "("A"+"B"i) / ("C"+"D"i) = ";
1480 !DIV,@A,@B,@C,@D
1490 PRINT A;
1500 IF B>=0 THEN PRINT"+";
1510 PRINT B"i"
1520 '
1530 END
1540 '
1550 '-----
1560 ' Donnees des RSX ADD, SCU, MUL et DIV
1570 '-----
1580 '
1590 DATA 0,0,0,0,12,90,C3,55,90,C3,A6,90,C3,3,91,C3
1600 DATA 8A,91,41,44,C4,53,4F,D5,4D,55,CC,44,49,D6,0,0
1610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1630 DATA 0,0,0,0,0,0,0,0,0,0,0,1,4,90,21,0
1640 DATA 90,CD,D1,BC,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66
1650 DATA 3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90
1660 DATA CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21
1670 DATA 23,90,11,2D,90,CD,7C,BD,21,23,90,ED,5B,1F,90,CD
1680 DATA 92,92,21,28,90,11,32,90,CD,7C,BD,21,28,90,ED,5B
1690 DATA 21,90,CD,92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD
1700 DATA 66,3,DD,6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21
1710 DATA 90,CD,53,92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92
1720 DATA 21,2D,90,CD,91,BD,21,23,90,11,2D,90,CD,7C,BD,21
1730 DATA 23,90,ED,5B,1F,90,CD,92,92,21,32,90,CD,91,BD,21
1740 DATA 28,90,11,32,90,CD,7C,BD,21,28,90,ED,5B,21,90,CD

```

```

1750 DATA 92,92,C9,DD,66,1,DD,6E,0,CD,65,92,DD,66,3,DD
1760 DATA 6E,2,CD,5C,92,DD,66,5,DD,6E,4,22,21,90,CD,53
1770 DATA 92,DD,66,7,DD,6E,6,22,1F,90,CD,4A,92,21,23,90
1780 DATA CD,77,92,21,3C,90,11,2D,90,CD,85,BD,21,28,90,CD
1790 DATA 80,92,21,41,90,11,32,90,CD,85,BD,21,41,90,CD,91
1800 DATA BD,21,3C,90,11,41,90,CD,7C,BD,21,3C,90,ED,5B,1F
1810 DATA 90,CD,92,92,21,23,90,11,32,90,CD,85,BD,21,28,90
1820 DATA 11,2D,90,CD,85,BD,21,23,90,11,28,90,CD,7C,BD,21
1830 DATA 23,90,ED,5B,21,90,CD,92,92,C9,DD,66,1,DD,6E,0
1840 DATA CD,65,92,DD,66,3,DD,6E,2,CD,5C,92,DD,66,5,DD
1850 DATA 6E,4,22,21,90,CD,53,92,DD,66,7,DD,6E,6,22,1F
1860 DATA 90,CD,4A,92,21,23,90,CD,77,92,21,3C,90,11,2D,90
1870 DATA CD,65,BD,21,28,90,CD,80,92,21,41,90,11,32,90,CD
1880 DATA 85,BD,21,3C,90,11,41,90,CD,7C,BD,21,2D,90,CD,80
1890 DATA 92,21,41,90,11,41,90,CD,85,BD,21,32,90,CD,89,92
1900 DATA 21,46,90,11,46,90,CD,85,BD,21,41,90,11,46,90,CD
1910 DATA 7C,BD,21,3C,90,11,41,90,CD,88,BD,21,3C,90,ED,5B
1920 DATA 1F,90,CD,92,92,21,23,90,11,32,90,CD,85,BD,21,23
1930 DATA 90,CD,91,BD,21,28,90,11,2D,90,CD,85,BD,21,23,90
1940 DATA 11,28,90,CD,7C,BD,21,23,90,11,41,90,CD,88,BD,21
1950 DATA 23,90,ED,5B,21,90,CD,92,92,C9,11,23,90,1,5,0
1960 DATA ED,B0,C9,11,28,90,1,5,0,ED,B0,C9,11,2D,90,1
1970 DATA 5,0,ED,B0,C9,11,32,90,1,5,0,ED,B0,C9,11,37
1980 DATA 90,1,5,0,ED,B0,C9,11,3C,90,1,5,0,ED,B0,C9
1990 DATA 11,41,90,1,5,0,ED,B0,C9,11,46,90,1,5,0,ED
2000 DATA 80,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0

```

Checksum :

```

63 B2 0 0 B6 52 7B EE 26 FD A3 51 5E C8 C1 37 6B BB 81 4D 3D EC 17 A7 FC 3F D8 A
4 60 49 47 99 56 A0 3C BE 36 70 F7 4B 2D E8

```

Dans ces programmes, les lignes 1130 à 1530 effectuent une démonstration de l'utilisation des RSX !ADD, !SOU, !MUL et !DIV. La ligne 1170 initialise les RSX par un appel à la routine DEFRSX :

```
1170 CALL &904B
```

Les lignes suivantes montrent comment passer des données aux RSX par l'intermédiaire de variables réelles.

Exemple :

En ligne 1250, les variables sont initialisées :

```
1250 A=10:B=5:C=-3:D=4.25
```


En ligne 1260, l'opération est affichée sur l'écran :

```
1260 PRINT ("A"+"B"i) + ("C"+"D"i) = "
```

La RSX !ADD est appelée ligne 1270 :

```
1270 !ADD, @A, @B, @C, @D
```

et le résultat est affiché lignes 1280 à 1300 :

```
1280 PRINT A;
1290 IF B>=0 THEN PRINT"+";
1300 PRINT B;
```

La ligne 1290 affiche le signe **+** entre la partie réelle et la partie imaginaire uniquement dans le cas où la partie imaginaire est positive. Cela permet d'éviter un résultat du type (par exemple) 4.7 +- 12i.

II. Changement de repère

Il est souvent utile de représenter un nombre complexe par son module et son argument :

$$a + ib = [f, \theta] = f (\cos \theta + i \sin \theta)$$

où f est appelé module et θ est appelé argument du nombre complexe.

Le module est égal à la racine carrée de $a^2 + b^2$:

$$f = \text{RAC}(a^2 + b^2)$$

L'équation

$$a + ib = f (\cos \theta + i \sin \theta)$$

permet de trouver la valeur de :

$$f \cos \theta = a$$

$$f \sin \theta = b$$

$$\text{d'où } \theta = \text{ArcCos}(a/f) = \text{ArcSin}(b/f)$$

Selon la valeur de a et de b , l'arc cosinus et l'arc sinus ne donneront pas forcément les mêmes valeurs.

Finalement :

- si a est positif,
= $\text{ArcTg}(b/a)$
- si a est négatif,
= $180 + \text{ArcTg}(b/a)$

Nous allons étudier la RSX !MA qui renvoie le module et l'argument du nombre complexe qui lui est passé.

COMMENT UTILISER LA RSX

Le listing de la RSX est le suivant

```

1          ORG  9000H
2          LOAD 9000H
3          ;-----
4          ; RSX MA
5          ; Calcul du module et de
6          ; l'argument d'un nombre complexe
7          ; Format : !MA,@x,@y
8          ; Entree (x)=Partie reelle
9          ;          (y)=Partie imaginaire
10         ; Sortie (x)=Module de x+iy
11         ;          (y)=Argument de x+iy
12         ;-----
13         ;
14         ;
15         ;-----
16         ; Declaration des constantes
17         ; et des variables du programme
18         ;-----
19         ;
20         ENTFL0:    EQU  0BD61H          ;Ent->Flottant
21         SIGFL0:    EQU  0BD91H          ;Signe flot
22         ADDFL0:    EQU  0BD79H          ;Addition flot
23         MULFL0:    EQU  0BD82H          ;Multiplie flot
24         DIVFL0:    EQU  0BD85H          ;Division flot
25         RACFL0:    EQU  0BD9AH          ;Rac flot
26         ATNFL0:    EQU  0BD82H          ;ATN flot
27         UNIT:      EQU  0BD94H          ;Unité de calcul
28         SAVHL:     DS    2              ;Sauvegarde de HL
29         SAVHL2:    DS    2              ;2eme sauv de HL
30         Z1:        DS    5              ;Zone reel 1
31         Z2:        DS    5              ;Zone reel 2

```

```

32          Z3:          DS    5          ;Zone reel 3
33          Z4:          DS    5          ;Zone reel 4
34          Z5:          DS    5          ;Zone reel 5
35          Z6:          DS    5          ;Zone reel 6
36          LOGEXT:     EQU    0BCD1H     ;KL LOG EXT
37          BUF:        DS    4          ;Zone RAM pour LOG EXT
38 9026 2B90          PTRTAB:     DW    TABLE     ;Pointeur TABLE
39 9028 C33890          JP    MA          ;Traitement du MA
40 902B 4D          TABLE:     DB    "M"
41 902C C1          DB    "A"+80H
42 902D 00          DB    0          ;Fin de table
43          ;
44          ;-----
45          ; Definition de la RSX
46          ;-----
47          ;
48          DEFRSX:     EQU    $          ;Point d'entree
49 902E 012690          LD    BC,PRTAB     ;Ptr table definition
50 9031 212290          LD    HL,BUF     ;Buffer pour LOG EXT
51 9034 CDD1BC          CALL LOGEXT     ;Definition de la RSX
52 9037 C9          RET
53          ;
54          ;-----
55          ; Traitement de ASIN
56          ;-----
57          ;
58          MA:         EQU    $          ;Point d'entree
59 9038 3EFF          LD    A,0FFH
60 903A CD94BD          CALL UNIT     ;Calcul en degres

```

```

61          ;
62 903D DD6601      LD  H, (IX+1)
63 9040 DD6E00      LD  L, (IX+0)          ; Adresse de la var. y
64 9043 220090      LD  (SAVHL),HL        ; Sauvegarde
65 9046 CDD890      CALL ZONE1          ; Memorisation
66 9049 DD6603      LD  H, (IX+3)
67 904C DD6E02      LD  L, (IX+2)          ; Adresse de la var. x
68 904F 220290      LD  (SAVHL2),HL       ; Sauvegarde
69 9052 CDE490      CALL ZONE2          ; Memorisation
70          ;
71 9055 210490      LD  HL, Z1
72 9058 110E90      LD  DE, Z3
73 905B CDED90      CALL FLODEHL        ; Sauvegarde
74 905E 210E90      LD  HL, Z3
75 9061 110E90      LD  DE, Z3
76 9064 CDB2B0      CALL MULFLO          ; y^2
77 9067 210990      LD  HL, Z2
78 906A 111390      LD  DE, Z4
79 906D CDED90      CALL FLODEHL        ; Sauvegarde
80 9070 211390      LD  HL, Z4
81 9073 111390      LD  DE, Z4
82 9076 CDB2B0      CALL MULFLO          ; x^2
83 9079 210E90      LD  HL, Z3
84 907C 111390      LD  DE, Z4
85 907F CD79B0      CALL ADDFLO          ; x^2+y^2
86 9082 210E90      LD  HL, Z3
87 9085 CD9AB0      CALL RACFLO          ; Rac (x^2+y^2)
88          ;
89 9088 210E90      LD  HL, Z3
90 908B EDSB0290     LD  DE, (SAVHL2)
91 908F CDED90      CALL FLODEHL        ; Module

```

```

92          ;
93 9092 210490      LD  HL,Z1
94 9095 111390      LD  DE,Z4
95 9098 CDED90      CALL FLODEHL      ;Sauvegarde
96 909B 210990      LD  HL,Z2
97 909E 111890      LD  DE,Z5
98 90A1 CDED90      CALL FLODEHL      ;Sauvegarde
99 90A4 211390      LD  HL,Z4
100 90A7 111890     LD  DE,Z5
101 90AA CDB5BD     CALL DIVFLO      ; y/x
102 90AD 211390     LD  HL,Z4
103 90B0 CDB2BD     CALL ATNFLO      ; ATN(y/x)
104 90B3 210990     LD  HL,Z2
105 90B6 CD91BD     CALL SIGFLO      ; x>0 ?
106 90B9 FEFF       CP   255          ; Non
107 90BB 2013       JR   NZ,POSITIF
108          ;
109 90BD AF          XOR  A           ;RAZ flag retenue
110 90BE 21B400     LD  HL,180
111 90C1 111D90     LD  DE,Z6
112 90C4 CD61BD     CALL ENTFLO      ;Z6=180 en flottant
113 90C7 211390     LD  HL,Z4
114 90CA 111D90     LD  DE,Z6
115 90CD CD79BD     CALL ADDFLO      ;180+ATN(y/x)
116          ;
117          POSITIF: EQU  *
118 90D0 211390     LD  HL,Z4
119 90D3 ED5B0090   LD  DE,(SAVHL)
120 90D7 CDED90     CALL FLODEHL      ; Argument

```

```
121          ;
122          ;
123          FIN:      EQU  $                ;Fin du programme
124 90DA C9          RET
125          ;
126          ;
127          ;-----
128          ; Zone des sous-programmes
129          ;-----
130          ;
131          ;-----
132          ; Transfert des BC octets pointes
133          ; par HL dans le buffer Z1
134          ;-----
135          ;
136          ZONE1:    EQU  $
137 90DB 110490      LD   DE,Z1
138 90DE 010500      LD   BC,5
139 90E1 EDB0        LDIR
140 90E3 C9          RET
141          ;
142          ;-----
143          ; Transfert des BC octets pointes
144          ; par HL dans le buffer Z2
145          ;-----
146          ;
147          ZONE2:    EQU  $
148 90E4 110990      LD   DE,Z2
149 90E7 010500      LD   BC,5
150 90EA EDB0        LDIR
151 90EC C9          RET
```

```

152          ;
153          ; -----
154          ; Transfert flottant de (HL)
155          ; dans (DE)
156          ; -----
157          ;
158          FLODEHL: EQU #
159 90ED 010500      LD  EC,5
160 90F0 EDB0       LDIR
161 90F2 09        RET
162                END

```

Cette RSX utilise des macros sur les nombres réels. Ainsi, il convient de modifier les adresses de ces macros en fonction de l'ordinateur utilisé (464, 664 ou 6128). Le listing a été écrit sur un CPC 664.

Ce tableau ci-dessous donne les correspondances pour les autres machines :

	CPC 464	CPC 664	CPC 6128
ENTFLO	BD40H	BD61H	BD64H
SIGFLO	BD70H	BD91H	BD94H
ADDFLO	BD58H	BD79H	BD7CH
MULFLO	BD61H	BD82H	BD85H
DIVFLO	BD64H	BD85H	BD88H
RACFLO	BD79H	BD9AH	BD9DH
ATNFLO	BD91H	BDB2H	BDB5H
UNIT	BD73H	BD94H	BD97H

Si vous préférez utiliser un chargeur Basic, voici les listings correspondants pour les trois versions de CPC :

Version CPC 464 :

```

1000 '=====
1010 ' Demonstration de la RSX MA
1020 '=====
1030 ' Version CPC 464
1040 '=====
1050 '
1060 FOR i=&9026 TO &90F2
1070   READ a$
1080   a$="&"+a$
1090   a=VAL(a$)
1100   POKE i,a
1110 NEXT i
1120 '
1130 '-----
1140 ' Initialisation de la RSX
1150 '-----
1160 '
1170 CALL &902E
1180 '
1190 '-----
1200 ' Demonstration
1210 '-----
1220 CLS
1230 PRINT"Utilisation de la RSX IMA"
1240 PRINT"-----"
1250 PRINT
1260 PRINT
1270 A=2:B=3
1280 PRINT "Le nombre complexe "A"+"B"i s'écrit également "
1290 !MA,@A,@B
1300 PRINT
1310 PRINT " ["A","B"]"
1320 PRINT
1330 PRINT "ou "A"est le module,"
1340 PRINT "et"B"l'argument du complexe."
1350 '
1360 END
1370 '
1380 '-----
1390 ' Donnees de la RSX MA
1400 '-----
1410 '
1420 DATA 2B,90,C3,38,90,4D,C1,0,1,26,90,21,22,90,CD,D1
1430 DATA BC,C9,3E,FF,CD,73,BD,DD,66,1,DD,6E,0,22,0,90
1440 DATA CD,DB,90,DD,66,3,DD,6E,2,22,2,90,CD,E4,90,21
1450 DATA 4,90,11,E,90,CD,ED,90,21,E,90,11,E,90,CD,61
1460 DATA BD,21,9,90,11,13,90,CD,ED,90,21,13,90,11,13,90

```



```

1470 DATA CD,61,BD,21,E,90,11,13,90,CD,58,BD,21,E,90,CD
1480 DATA 79,BD,21,E,90,ED,5B,2,90,CD,ED,90,21,4,90,11
1490 DATA 13,90,CD,ED,90,21,9,90,11,18,90,CD,ED,90,21,13
1500 DATA 90,11,18,90,CD,64,BD,21,13,90,CD,91,BD,21,9,90
1510 DATA CD,70,BD,FE,FF,20,13,AF,21,B4,0,11,1D,90,CD,40
1520 DATA BD,21,13,90,11,1D,90,CD,58,BD,21,13,90,ED,5B,0
1530 DATA 90,CD,ED,90,C9,11,4,90,1,5,0,ED,B0,C9,11,9
1540 DATA 90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0

```

Checksum :

```
82 8 EB 2F F2 D2 E5 E4 D6 80 33 D4 6D
```

Version CPC 664 :

```

1000 '=====  

1010 ' Demonstration de la RSX MA  

1020 '=====  

1030 ' Version CPC 664  

1040 '=====  

1050 '  

1060 FOR i=&9026 TO &90F2  

1070   READ a$  

1080   a$="&"+a$  

1090   a=VAL(a$)  

1100   POKE i,a  

1110 NEXT i  

1120 '  

1130 '-----  

1140 ' Initialisation de la RSX  

1150 '-----  

1160 '  

1170 CALL &902E  

1180 '  

1190 '-----  

1200 ' Demonstration  

1210 '-----  

1220 CLS  

1230 PRINT"Utilisation de la RSX IMA"  

1240 PRINT"-----"  

1250 PRINT  

1260 PRINT  

1270 A=2:B=3  

1280 PRINT "Le nombre complexe "A"+"B"i s'ecrit egalement "  

1290 IMA,@A,@B  

1300 PRINT  

1310 PRINT " ["A","B"]"  

1320 PRINT  

1330 PRINT "ou "A"est le module,"  

1340 PRINT "et"B"l'argument du complexe."  

1350 '

```

```

1360 END
1370 '
1380 '-----
1390 ' Donnees de la RSX MA
1400 '-----
1410 '
1420 DATA 2B,90,C3,38,90,4D,C1,0,1,26,90,21,22,90,CD,D1
1430 DATA BC,C9,3E,FF,CD,94,BD,DD,66,1,DD,6E,0,22,0,90
1440 DATA CD,DB,90,DD,66,3,DD,6E,2,22,2,90,CD,E4,90,21
1450 DATA 4,90,11,E,90,CD,ED,90,21,E,90,11,E,90,CD,82
1460 DATA BD,21,9,90,11,13,90,CD,ED,90,21,13,90,11,13,90
1470 DATA CD,82,BD,21,E,90,11,13,90,CD,79,BD,21,E,90,CD
1480 DATA 9A,BD,21,E,90,ED,5B,2,90,CD,ED,90,21,4,90,11
1490 DATA 13,90,CD,ED,90,21,9,90,11,18,90,CD,ED,90,21,13
1500 DATA 90,11,18,90,CD,85,BD,21,13,90,CD,B2,BD,21,9,90
1510 DATA CD,91,BD,FE,FF,20,13,AF,21,B4,0,11,1D,90,CD,61
1520 DATA BD,21,13,90,11,1D,90,CD,79,BD,21,13,90,ED,5B,0
1530 DATA 90,CD,ED,90,C9,11,4,90,1,5,0,ED,B0,C9,11,9
1540 DATA 90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0

```

Checksum :

```
82 29 E8 50 F2 15 7 E4 19 C2 54 D4 60
```

Version CPC 6128 :

```

1000 '=====
1010 ' Demonstration de la RSX MA
1020 '=====
1030 ' Version CPC 6128
1040 '=====
1050 '
1060 FOR i=&9026 TO &90F2
1070   READ a$
1080   a$="&"+a$
1090   a=VAL(a$)
1100   POKE i,a
1110 NEXT i
1120 '
1130 '-----
1140 ' Initialisation de la RSX
1150 '-----
1160 '
1170 CALL &902E
1180 '
1190 '-----
1200 ' Demonstration
1210 '-----
1220 CLS
1230 PRINT"Utilisation de la RSX IMA"
1240 PRINT"-----"

```

```

1250 PRINT
1260 PRINT
1270 A=2:B=3
1280 PRINT "Le nombre complexe "A"+"B"i s'écrit également "
1290 !MA,@A,@B
1300 PRINT
1310 PRINT " ["A","B"]"
1320 PRINT
1330 PRINT "ou "A"est le module,"
1340 PRINT "et"B"l'argument du complexe."
1350 '
1360 END
1370 '
1380 '-----
1390 ' Donnees de la RSX MA
1400 '-----
1410 '
1420 DATA 2B,90,C3,3B,90,4D,C1,0,1,26,90,21,22,90,CD,D1
1430 DATA BC,C9,3E,FF,CD,97,BD,DD,66,1,DD,6E,0,22,0,90
1440 DATA CD,DB,90,DD,66,3,DD,6E,2,22,2,90,CD,E4,90,21
1450 DATA 4,90,11,E,90,CD,ED,90,21,E,90,11,E,90,CD,85
1460 DATA BD,21,9,90,11,13,90,CD,ED,90,21,13,90,11,13,90
1470 DATA CD,85,BD,21,E,90,11,13,90,CD,7C,BD,21,E,90,CD
1480 DATA 9D,BD,21,E,90,ED,5B,2,90,CD,ED,90,21,4,90,11
1490 DATA 13,90,CD,ED,90,21,9,90,11,18,90,CD,ED,90,21,13
1500 DATA 90,11,18,90,CD,8B,BD,21,13,90,CD,B5,BD,21,9,90
1510 DATA CD,94,BD,FE,FF,20,13,AF,21,B4,0,11,1D,90,CD,64
1520 DATA BD,21,13,90,11,1D,90,CD,7C,BD,21,13,90,ED,5B,0
1530 DATA 90,CD,ED,90,C9,11,4,90,1,5,0,ED,B0,C9,11,9
1540 DATA 90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0

```

Checksum :

82 2C E8 53 F2 1B A E4 1F C8 57 D4 6D

Si vous utilisez la version assembleur, n'oubliez pas d'installer la RSX à l'aide d'un CALL &902E.

Le format d'appel de la RSX est le suivant :

!MA, @R, @I

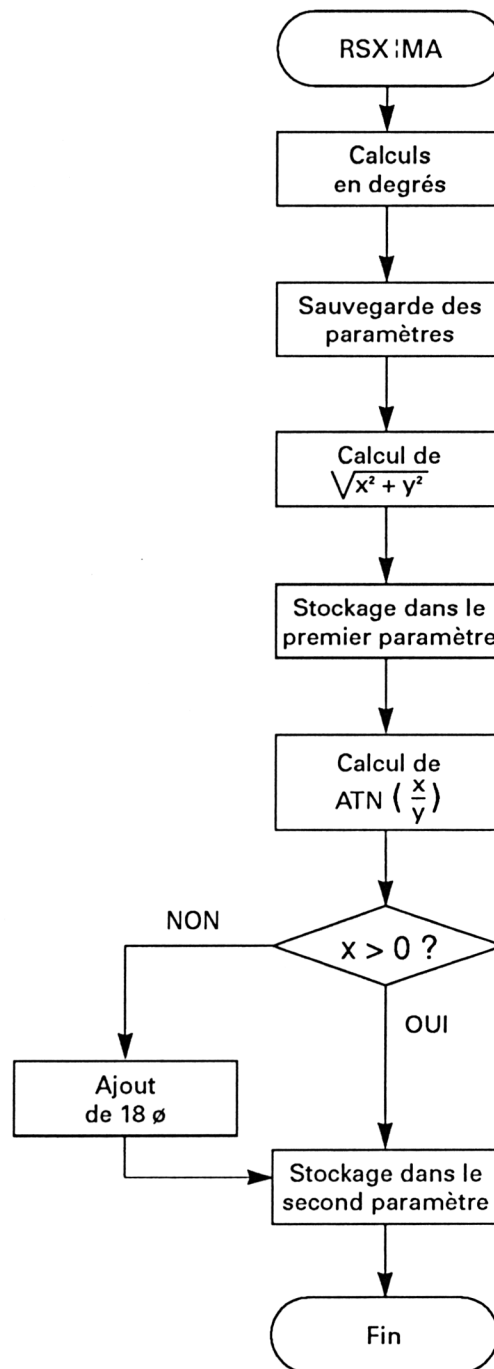
où R est une variable réelle qui contient la partie réelle du complexe, et I est une variable réelle qui contient la partie imaginaire du complexe.

Le résultat est renvoyé dans les variables R et I :

- la variable R contient le module ;
- la variable I contient l'argument.

LA RSX EN DÉTAIL

La logique de fonctionnement de la RSX apparaît dans l'ordinogramme suivant :



Les premières lignes du programme de traitement demandent à l'Amstrad de travailler en degrés à l'aide d'un appel à la macro UNIT :

```
MA: EQU $
      LD A,0FFH
      CALL UNIT
```

Les adresses des deux paramètres passés à la RSX sont mémorisés dans les variables SAVHL et SAVHL2. Leurs valeurs réelles sont stockées dans les variables Z1 et Z2.

```
LD H,(IX+1)
LD L,(IX+0)
LD (SAVHL),HL —————> Mémorisation de l'adresse
CALL ZONE1 —————> Mémorisation de la valeur réelle
LD H,(IX+3)
LD L,(IX+2)
LD (SAVHL2),HL —————> Mémorisation de l'adresse
CALL ZONE2 —————> Mémorisation de la valeur réelle
```

Le module étant calculé par la racine carrée de la somme des carrés des parties réelle et imaginaire du complexe, le programme calcule ensuite les valeurs y^2 (lignes 71 à 76), x^2 (lignes 77 à 82), la somme $x^2 + y^2$ (lignes 83 à 85) et extrait la racine carrée de ce nombre (lignes 86 et 87).

Le résultat est stocké dans le premier argument lignes 89 à 91.

Le programme se poursuit par le calcul de l'argument.

La valeur $\text{ANT}(y/x)$ est calculée lignes 99 à 103. Si cette valeur est positive, elle représente l'argument. Le test lignes 104 à 107 donne alors le contrôle à la ligne d'étiquette POSITIF.

Dans le cas contraire, il faut ajouter 180 à cette valeur pour obtenir l'argument.

La valeur entière 180 est convertie en un réel lignes 109 à 112. L'instruction XOR A ligne 109 est très importante. Elle met à zéro l'indicateur de retenue qui pourrait fausser la conversion. L'addition se fait lignes 113 à 115.

L'argument est stocké dans le second argument lignes 118 à 120.

Le programme se termine par trois sous-programmes fort utiles.

- ZONE1 transfère les BC octets pointés par HL dans la zone mémoire réelle Z1 ;
- ZONE2 transfère les BC octets pointés par HL dans la zone réelle Z2 ;
- FLODEHL transfère la donnée réelle pointée par HL dans les mémoires pointées par DE.

III. Racine d'un nombre complexe

Pour compléter votre bibliothèque de RSX dédiées à la manipulation de nombres complexes, nous vous proposons la RSX **RAC** qui extrait la racine d'un nombre complexe exprimé sous forme algébrique.

Un peu de théorie.

Un nombre complexe Z exprimé sous forme algébrique se présente comme suit :

$$Z = a + ib$$

Comment trouver la partie réelle et la partie imaginaire d'un complexe Z' tel que :

$$Z'^2 = Z$$

Posons :

$$Z' = c + id$$

Il vient

$$Z'^2 = Z$$

$$\Leftrightarrow (c+id)^2 = a+ib$$

$$\Leftrightarrow c^2-d^2 = a$$

$$\text{et } 2cd = b$$

$$\text{Or, } |(c+id)^2| = |c+id|^2$$

et comme $(c+id)^2 = a + ib$, il vient

$$\text{donc, } \sqrt{a^2+b^2} = c^2+d^2$$

Ajoutons cette équation aux deux précédentes. Il vient :

$$\begin{cases} c^2-d^2 = a \\ c^2+d^2 = \sqrt{a^2+b^2} \\ 2cd = b \end{cases}$$

Si nous ajoutons la première et la seconde équation, nous obtenons :

$$2c^2 = a + \sqrt{a^2+b^2}$$

$$\text{d'où } c = \sqrt{(a+\sqrt{a^2+b^2})/2}$$

De même, en soustrayant la première équation de la seconde, il vient :

$$2d^2 = \sqrt{a^2+b^2} - a$$

$$\text{d'où, } d = \sqrt{(\sqrt{a^2+b^2} - a)/2}$$

Ces deux formules seront utilisées dans la RSX pour calculer le module (c) et l'argument (d) de la racine.

COMMENT UTILISER LE PROGRAMME

Le programme est présenté sous la forme d'une RSX écrite en assembleur et sous la forme d'un chargeur Basic.

Si vous désirez utiliser la RSX, saisissez le listing suivant :

```

1          ORG  9000H
2          LOAD 9000H
3          ;-----
4          ; RSX RAC
5          ; Format : IRAC,r,i
6          ; Entree : r=partie reelle argument
7          ;          i=partie imagin argument
8          ; Sortie : r=partie reelle resultat
9          ;          i=partie imagin resultat
10         ;-----
11         ;
12         ;
13         ;-----
14         ; Declaration des constantes
15         ; et des variables du programme
16         ;-----
17         ;
18         ENTFLO:   EQU  0BD61H           ;Entier->Flot
19         ADDFLO:   EQU  0BD79H           ;Addition flot
20         NEGFLO:   EQU  0BD8EH           ;Negation flot
21         MULFLO:   EQU  0BD82H           ;Multipl flot
22         DIVFLO:   EQU  0BD85H           ;Division flot
23         RACFLO:   EQU  0BD9AH           ;Racine flot

```

```

24      LOGEXT:      EQU  0BCD1H          ;KL LOG EXT
25      ;
26      BUF:         DS    4              ;ZONE RAM POUR LOG EXT
27 9004 0990      PTRTAB:      DW    TABLE      ;Pointeur TABLE
28 9006 C33490      JP    RAC              ;Calcul de la racine
29 9009 5241      TABLE:      DB    "RA"
30 900B C3         DB    "C"+80H
31 900C 00         DB    0                ;Fin de table
32      ;
33      SAVRE:       DS    2              ;Adresse partie reelle
34      SAVIM:       DS    2              ;Adresse partie imagin
35      Z1:          DS    5              ;Zone reel 1
36      Z2:          DS    5              ;Zone reel 2
37      Z3:          DS    5              ;Zone reel 3
38      Z4:          DS    5              ;Zone reel 4
39      Z5:          DS    5              ;Zone reel 5
40      ;
41      ;-----
42      ; Definition de la RSX
43      ;-----
44      ;
45      DEFRSX:      EQU    $              ;Point d'entree
46 902A 010490      LD    BC,PRTAB        ;Ptr table definition
47 902D 210090      LD    HL,BUF         ;Buffer pour LOG EXT
48 9030 CDD1BC      CALL LOGEXT          ;Definition de la RSX

```



```

49 9033 C9          RET
50                ;
51                ;
52                ;-----
53                ; Traitement de RAC
54                ;-----
55                ;
56                RAC:      EQU $          ;Point d'entree
57                ;
58                ;-----
59                ; Lecture des donnees passees
60                ;-----
61                ;
62 9034 DD6601      LD   H,(IX+1)
63 9037 DD6E00      LD   L,(IX+0)
64 903A 220F90      LD   (SAVIM),HL      ;Sauvegarde imagin
65 903D CDDE90      CALL ZONE2          ;Memo 2eme argument
66 9040 DD6603      LD   H,(IX+3)
67 9043 DD6E02      LD   L,(IX+2)
68 9046 220D90      LD   (SAVRE),HL      ;Sauvegarde reelle
69 9049 CDD590      CALL ZONE1          ;Memo 1er argument
70 904C 211190      LD   HL,Z1
71 904F 111B90      LD   DE,Z3
72 9052 CDE790      CALL FLODEHL        ;Sauvegarde

```

```

73          ;
74          ;-----
75          ; Calcul de la racine
76          ;-----
77          ;
78 9055 211B90      LD   HL,Z3
79 9058 111B90      LD   DE,Z3
80 905B CD82BD      CALL MULFLO      ;Calcul de C^2
81 905E 211690      LD   HL,Z2
82 9061 111690      LD   DE,Z2
83 9064 CD82BD      CALL MULFLO      ;Calcul de D^2
84 9067 211690      LD   HL,Z2
85 906A 111B90      LD   DE,Z3
86 906D CD79BD      CALL ADDFLO      ;C^2 + D^2
87 9070 211690      LD   HL,Z2
88 9073 CD9ABD      CALL RACFLO      ;RAC(C^2+D^2)
89 9076 211690      LD   HL,Z2
90 9079 112590      LD   DE,Z3
91 907C CDE790      CALL FLODEHL     ;Sauvegarde
92 907F 211690      LD   HL,Z2
93 9082 111190      LD   DE,Z1
94 9085 CD79BD      CALL ADDFLO      ;RAC(C^2+D^2)+C
95 9088 3F          SCF
96 9089 3F          CCF

```

```

97 908A 210200      LD   HL,2
98 908D 112090      LD   DE,Z4
99 9090 CD61BD      CALL ENTFL0           ;2 en flottant
100 9093 211690     LD   HL,Z2
101 9096 112090     LD   DE,Z4
102 9099 CD85BD     CALL DIVFL0           ; (RAC(C^2+D^2)+C)/2
103 909C 211690     LD   HL,Z2
104 909F CD9ABD     CALL RACFL0           ;RAC((RAC(C^2+D^2)+C)/2)
105 90A2 211690     LD   HL,Z2
106 90A5 ED5B0D90   LD   DE,(SAVRE)
107 90A9 CDE790     CALL FLODEHL          ;Partie reelle resultat
108 90AC 211190     LD   HL,Z1
109 90AF CD8EBD     CALL NEGFL0           ;-C
110 90B2 212590     LD   HL,Z5
111 90B5 111190     LD   DE,Z1
112 90B8 CD79BD     CALL ADDFL0           ;RAC(C^2+D^2)-C
113 90BB 212590     LD   HL,Z5
114 90BE 112090     LD   DE,Z4
115 90C1 CD85BD     CALL DIVFL0           ; (RAC(C^2+D^2)-C)/2
116 90C4 212590     LD   HL,Z5
117 90C7 CD9ABD     CALL RACFL0           ;RAC((RAC(C^2+D^2)-C)/2)
118 90CA 212590     LD   HL,Z5
119 90CD ED5B0F90   LD   DE,(SAVIM)
120 90D1 CDE790     CALL FLODEHL          ;Partie imagin resultat

```

```
121 90D4 C9          RET
122                ;
123                ;-----
124                ; Zone des sous-programmes
125                ;-----
126                ;
127                ;-----
128                ; Transfert des BC octets pointes
129                ; par HL dans Z1, Z4 ou Z5
130                ;-----
131                ;
132                ZONE1:    EQU  $
133 90D5 111190      LD  DE,Z1
134 90D8 010500      LD  BC,5
135 90DB EDB0        LDIR
136 90DD C9          RET
137                ;
138                ZONE2:    EQU  $
139 90DE 111690      LD  DE,Z2
140 90E1 010500      LD  BC,5
141 90E4 EDB0        LDIR
142 90E6 C9          RET
143                ;
144                ;
```

```

145      ;-----
146      ; Transfert flottant de (HL)
147      ; dans (DE)
148      ;-----
149      ;
150      FLODEHL:   EQU   #
151      90E7 010500      LD   BC,5
152      90EA EDB0      LDIR
153      90EC C9      RET
154      END

```

Compilez le programme et activez la RSX en tapant sous Basic :
CALL &902A

La RSX **RAC** est désormais prête à fonctionner. Vous devez lui passer deux paramètres par adresse :

- la partie réelle de l'argument ;
- la partie imaginaire de l'argument.

Le résultat est renvoyé dans les adresses passées.

Exemple :

Cherchons la racine du nombre complexe $4 + 3i$.

Pour cela, il suffit de taper :

A=4 : B=3 :| RAC, @A, @B : PRINT A, B

Le résultat affiché est

2.12132034 0.707106782

Effectivement, on peut vérifier que

$(2.12132034 + 0.707106782 i)^2 = 4 + 3i$

Si vous préférez utiliser un chargeur Basic, tapez le listing suivant :

```

1000 REM -----
1010 REM Chargeur BASIC de la RSX IRAC
1020 REM -----
1030 REM
1040 FOR i=&9000 TO &90EC
1050   READ a$
1060   a$="&"+a$
1070   POKE i,VAL(a$)
1080 NEXT i
1090 CALL &902A
1100 END
1110 REM - - - - -
1120 REM Donnees de la RSX
1130 REM - - - - -
1140 REM
1150 DATA FC,A6,4,90,9,90,C3,34,90,52,41,C3,0,AA,1,B3
1160 DATA 1,0,0,0,80,83,68,B6,C3,7,82,0,0,0,0,85
1170 DATA 0,0,0,0,82,37,F3,4,35,80,1,4,90,21,0,90
1180 DATA CD,D1,BC,C9,DD,66,1,DD,6E,0,22,F,90,CD,DE,90
1190 DATA DD,66,3,DD,6E,2,22,D,90,CD,D5,90,21,11,90,11
1200 DATA 1B,90,CD,E7,90,21,1B,90,11,1B,90,CD,82,BD,21,16
1210 DATA 90,11,16,90,CD,82,BD,21,16,90,11,1B,90,CD,79,BD
1220 DATA 21,16,90,CD,9A,BD,21,16,90,11,25,90,CD,E7,90,21
1230 DATA 16,90,11,11,90,CD,79,BD,37,3F,21,2,0,11,20,90
1240 DATA CD,61,BD,21,16,90,11,20,90,CD,85,BD,21,16,90,CD
1250 DATA 9A,BD,21,16,90,ED,5B,D,90,CD,E7,90,21,11,90,CD
1260 DATA 8E,BD,21,25,90,11,11,90,CD,79,BD,21,25,90,11,20
1270 DATA 90,CD,85,BD,21,25,90,CD,9A,BD,21,25,90,ED,5B,F
1280 DATA 90,CD,E7,90,C9,11,11,90,1,5,0,ED,B0,C9,11,16
1290 DATA 90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0

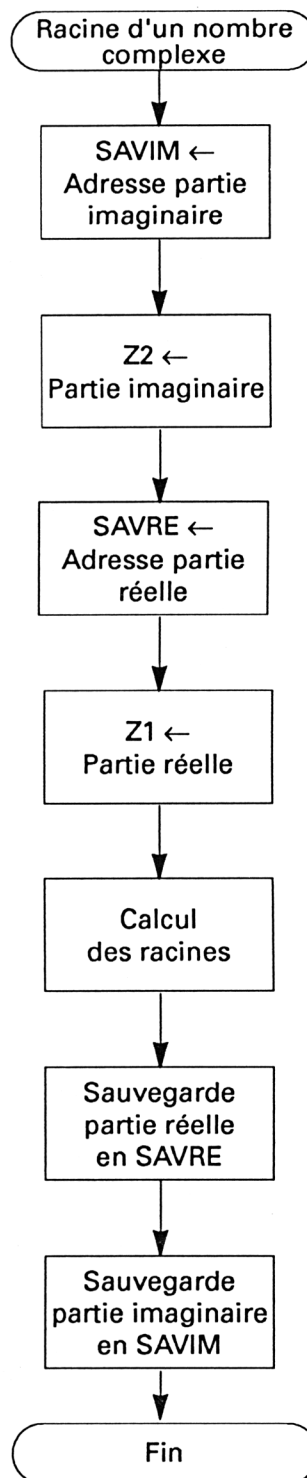
```

Vérifiez la conformité des lignes de DATA en utilisant le calculateur de checksum qui doit donner le résultat suivant :

```
11 F6 AE B6 5D C0 DF E3 B9 1D DD E2 CD EB 6D
```

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme ci-dessous :



Comme toujours, les premières lignes du listing sont utilisées pour déclarer les constantes et les variables du programme.

Signalons en particulier :

- les constantes XXXFLO (lignes 18 à 23) qui donnent accès aux points d'entrée des routines de manipulation de nombres réels ;
- la constante LOGEXT (ligne 24) qui permet de définir la RSX ;
- les zones BUF, PTRTAB et TABLE (lignes 26 à 31) qui contiennent toutes les données relatives à la RSX : nom et point d'entrée ;
- les zones Z1 à Z5 de 5 octets (lignes 33 à 39) qui permettent le stockage d'un nombre réel.

Le programme de définition de la RSX est des plus classiques. Il se trouve entre les lignes 45 et 49. Reportez-vous à la description d'une autre RSX si vous désirez plus de détail à son sujet.

Le programme se poursuit par le point d'entrée de la RSX (RAC ligne 56). Les instructions situées en-dessous de ce point d'entrée seront exécutées à chaque fois que le système d'exploitation de l'Amstrad détectera une instruction ;RAC. La première action effectuée par la routine de traitement consiste à sauver les deux paramètres passés :

- l'adresse de la partie imaginaire dans la zone SAVIM

```
LD    H, (IX+1)
LD    L, (IX+0)
LD    (SAVIM), HL
```

- la partie imaginaire dans la zone réelle Z2

```
CALL ZONE2
```

- l'adresse de la partie réelle dans la zone SAVRE

```
LD    H, (IX+3)
LD    L, (IX+2)
LD    (SAVRE), HL
```

- la partie réelle dans la zone réelle Z1

```
CALL ZONE1
```

La suite du programme calcule les quantités

$$c = \sqrt{(a + \sqrt{a^2 + b^2}) / 2}$$

$$\text{et } d = \sqrt{(\sqrt{a^2 + b^2} - a) / 2}$$

Nous n'entrerons pas dans le détail des calculs. En suivant le listing commenté de la RSX, vous pourrez découvrir par vous-même leur cheminement.

Nous préciserons seulement que le résultat des procédures **XXX-FLO** est toujours stocké dans la zone pointée par la paire de registres **HL**.

Par exemple, les instructions suivantes additionnent les nombres réels qui se trouvent dans les zones **Z2** et **Z3** et stockent le résultat dans la zone pointée par **HL** : **Z2**.

```
LD    HL, Z2
LD    DE, Z3
CALL  ADDFLO --> Z2+Z3 → Z2
```

Le programme se termine par la zone des sous-programmes lignes 133 à 153.

Le sous-programmes **ZONE1** et **ZONE2** transfèrent les **BC** octets pointés par la paire de registres **HL** dans les zones (respectivement) **Z1** et **Z2**. Pour cela, ils utilisent une instruction très puissante : **LDIR**.

Cette instruction recopie l'octet pointé par **HL** dans la mémoire pointée par **DE**. Elle décrémente ensuite **BC**, incrémente **HL** et **DE**. Le processus de recopie se poursuit jusqu'à ce que la paire de registres **BC** ait une valeur nulle.

Enfin, le sous-programme **FLODEHL** recopie le nombre réel pointé par **HL** dans la zone pointée par **DE**. Ce sous-programme utilise également une instruction **LDIR**.

