9/4.4

Bibliothèque mathématique

Dans les paragraphes qui suivent, nous allons agrandir la bibliothèque des fonctions mathématiques des AMSTRAD à l'aide de RSX. Dans un premier temps, nous allons étudier les RSX :

|ACOS → arc cosinus,
|ASIN → arc sinus,
|COSH → cosinus hyperbolique

Ces RSX mettent en œuvre les vecteurs mathématiques opérant sur les réels. Pour toutes informations complémentaires concernant les vecteurs ADDFLO, MULFLO, DIVFLO, NEGFLO, etc., vous pouvez vous reporter au chapitre *Accès aux vecteurs mathématiques en Assembleur*, Partie 4, chapitre 2.10.

9/4.4.1

RSX ACOS : arc cosinus

La fonction **arc cosinus** ne fait pas partie de la bibliothèque standard des CPC. Il est cependant possible de la simuler en utilisant une formule équivalente composée de fonctions qui font partie de la bibliothèque standard des CPC. En effet :

ACOS(X) = ATAN (SQR (1 – X^2) / X)

COMMENT EXÉCUTER LE PROGRAMME

Le programme mettant en œuvre une RSX, il est forcément écrit en Assembleur. Si vous désirez l'utiliser sous sa forme Assembleur, entrez le listing de la page suivante.

```
1                       ORG   9000H
2                       LOAD 9000H
3                 ;-------------------------------
4                 ; RSX ARC COSINUS
5                 ; Format : !ACOS,@VAR
6                 ; Entree (VAR)=Arc de l'angle
7                 ; Sortie (VAR)=Angle de cos (VAR)
8                 ;-------------------------------
9                 ;
10                ;
11                ;-------------------------------
12                ; Declaration des constantes
13                ; et des variables du programme
14                ;-------------------------------
15                ;
16                ENTFLO:     EQU   0BD61H          ;Ent->Flottant
17                COMFLO:     EQU   0BD8BH          ;Comp 2 flot
18                SIGFLO:     EQU   0BD91H          ;Signe flot
19                NEGFLO:     EQU   0BD8EH          ;Negation flot
20                ADDFLO:     EQU   0BD79H          ;Addition flot
21                SOUFLO:     EQU   0BD7FH          ;Soustr flot
22                MULFLO:     EQU   0BD82H          ;Multiplic flot
23                DIVFLO:     EQU   0BD85H          ;Division flot
24                RACFLO:     EQU   0BD9AH          ;Rac flot
25                PUIFLO:     EQU   0BD9DH          ;Puis flot
26                ATNFLO:     EQU   0BDB2H          ;ATN flot
27                UNIT:       EQU   0BD94H          ;Unite de calcul
28                SAVHL:      DS    2               ;Sauvegarde de HL
29                Z1:         DS    5               ;Zone reel 1
30                Z2:         DS    5               ;Zone reel 2
```

```
31                  Z3:         DS   5                ;Zone reel 3
32                  LOGEXT:     EQU  0BCD1H           ;KL LOG EXT
33                  BUF:        DS   4                ;Zone RAM pour LOG EXT
34 9015 1A90        PTRTAB:     DW   TABLE            ;Pointeur TABLE
35 9017 C32990                  JP   ACOS             ;Traitement du ACOS
36 901A 41434F      TABLE:      DB   "ACO"
37 901D D3                      DB   "S"+80H
38 901E 00                      DB   0                ;Fin de table
39                  ;
40                  ;---------------------------------
41                  ; Definition de la RSX
42                  ;---------------------------------
43                  ;
44                  DEFRSX:     EQU  $                ;Point d'entree
45 901F 011590                  LD   BC,PTRTAB        ;Ptr table definition
46 9022 211190                  LD   HL,BUF           ;Buffer pour LOG EXT
47 9025 CDD1BC                  CALL LOGEXT           ;Definition de la RSX
48 9028 C9                      RET
49                  ;
50                  ;---------------------------------
51                  ; Traitement de ACOS
52                  ;---------------------------------
53                  ;
54                  ACOS:       EQU  $                ;Point d'entree
55 9029 3EFF                    LD   A,0FFH
56 902B CD94BD                  CALL UNIT             ;Calcul en degres
57                  ;
58 902E DD6601                  LD   H,(IX+1)
59 9031 DD6E00                  LD   L,(IX+0)         ;Adresse de la var.
60 9034 220090                  LD   (SAVHL).HL       ;Sauvegarde
```

```
61 9037 CDE990              CALL  ZONE1           ;Memorisation
62 903A 210290              LD    HL,Z1
63 903D CDF290              CALL  ZONE3           ;Sauvegarde
64 9040 210290              LD    HL,Z1
65 9043 CD91BD              CALL  SIGFLO          ;Signe du param
66 9046 B7                  OR    A
67 9047 CAD190              JP    Z,ANGLE90
68 904A FEFF                CP    255
69 904C 2006                JR    NZ,SUITE        ;Nombre positif
70 904E 210C90              LD    HL,Z3
71 9051 CD8EBD              CALL  NEGFLO          ;Negation param
72              SUITE:      EQU   $
73 9054 210100              LD    HL,1
74 9057 110790              LD    DE,Z2
75 905A CD61BD              CALL  ENTFLO          ;1 en flottant
76 905D 210C90              LD    HL,Z3
77 9060 110790              LD    DE,Z2
78 9063 CD8BBD              CALL  COMFLO          ;Valeur de param
79 9066 FE01                CP    1               ;Param>1 ?
80 9068 2873                JR    Z,ANGLE255      ;Parametre incorrect
81                          ;
82 906A 210290              LD    HL,Z1
83 906D CDF290              CALL  ZONE3           ;Copie Z1 -> Z3
84 9070 210C90              LD    HL,Z3
85 9073 110290              LD    DE,Z1
86 9076 CD82BD              CALL  MULFLO          ;X * X
87 9079 210C90              LD    HL,Z3
88 907C CD8EBD              CALL  NEGFLO          ;- X * X
89                          ;
90 907F AF                  XOR   A               ;RAZ Flag retenue
```

```
 91 9080 210100                 LD    HL,1
 92 9083 110790                 LD    DE,Z2
 93 9086 CD61BD                 CALL  ENTFLO          ;1 en flottant
 94 9089 210790                 LD    HL,Z2
 95 908C 110C90                 LD    DE,Z3
 96 908F CD79BD                 CALL  ADDFLO          ;1 - X**2
 97 9092 210790                 LD    HL,Z2
 98 9095 CD9ABD                 CALL  RACFLO          ; RAC(1-X**2)
 99 9098 210790                 LD    HL,Z2
100 909B 110290                 LD    DE,Z1
101 909E CD85BD                 CALL  DIVFLO          ;RAC(1-X**2)/X
102 90A1 210790                 LD    HL,Z2
103 90A4 CDB2BD                 CALL  ATNFLO          ;ATN(RAC(1-X**2)/X)
104                 ;
105 90A7 210290                 LD    HL,Z1
106 90AA CD91BD                 CALL  SIGFLO
107 90AD FEFF                   CP    255
108 90AF 280B                   JR    Z,PLUS180       ;Angle+180
109                 ;
110                 SAVRES:     EQU   $
111 90B1 210790                 LD    HL,Z2
112 90B4 ED5B0090               LD    DE,(SAVHL)
113 90B8 CDFB90                 CALL  FLODEHL         ;Resultat
114                 ;
115                 FIN:        EQU   $               ;Fin du programme
116 90BB C9                     RET
117                 ;
118                 PLUS180:    EQU   $               ;Angle + 180
119 90BC AF                     XOR   A
120 90BD 21B400                 LD    HL,180
```

```
121 90C0 110C90              LD    DE,Z3
122 90C3 CD61BD              CALL  ENTFLO          ;180 en flottant
123 90C6 210790              LD    HL,Z2
124 90C9 110C90              LD    DE,Z3
125 90CC CD7FBD              CALL  SOUFLO          ;Resultat+180
126 90CF 18E0                JR    SAVRES
127                ;
128                ANGLE90:  EQU   $             ;Resultat=90
129 90D1 215A00              LD    HL,90
130 90D4 ED5B0090            LD    DE,(SAVHL)      ;Adresse du resultat
131 90D8 CD61BD              CALL  ENTFLO          ;Convers 90 en flot
132 90DB 18DE                JR    FIN
133                ;
134                ANGLE255: EQU   $             ;Parametre errone
135 90DD 21FF00              LD    HL,255
136 90E0 ED5B0090            LD    DE,(SAVHL)      ;Adresse du resultat
137 90E4 CD61BD              CALL  ENTFLO          ;Conversion/Stockage
138 90E7 18D2                JR    FIN
139                ;
140                ;-------------------------------
141                ; Zone des sous-programmes
142                ;-------------------------------
143                ;
144                ;-------------------------------
145                ; Transfert des BC octets pointes
146                ; par HL dans le buffer Z1
147                ;-------------------------------
148                ;
149                ZONE1:    EQU   $
150 90E9 110290              LD    DE.Z1
```

```
151 90EC 010500             LD    BC,5
152 90EF EDB0               LDIR
153 90F1 C9                 RET
154              ;
155              ;-------------------------------
156              ; Transfert des BC octets pointes
157              ; par HL dans le buffer Z3
158              ;-------------------------------
159              ;
160              ZONE3:      EQU   $
161 90F2 110C90             LD    DE,Z3
162 90F5 010500             LD    BC,5
163 90F8 EDB0               LDIR
164 90FA C9                 RET
165              ;
166              ;-------------------------------
167              ; Transfert flottant de (HL)
168              ; dans (DE)
169              ;-------------------------------
170              ;
171              FLODEHL:    EQU   $
172 90FB 010500             LD    BC,5
173 90FE EDB0               LDIR
174 9100 C9                 RET
175                         END
```

```
ADDFLO    BD79 ATNFLO    BDB2 ACOS      9029 ANGLE90   90D1
ANGLE255  90DD BUF       9011 COMFLO    BD8B DIVFLO    BD85
DEFRSX    901F ENTFLO    BD61 FIN       90BB FLODEHL   90FB
LOGEXT    BCD1 MULFLO    BD82 NEGFLO    BD8E PUIFLO    BD9D
PTRTAB    9015 PLUS180   90BC RACFLO    BD9A SIGFLO    BD91
SOUFLO    BD7F SAVHL     9000 SUITE     9054 SAVRES    90B1
TABLE     901A UNIT      BD94 Z1        9002 Z2        9007
Z3        900C ZONE1     90E9 ZONE3     90F2
```

Installez la RSX en tapant sous Basic :

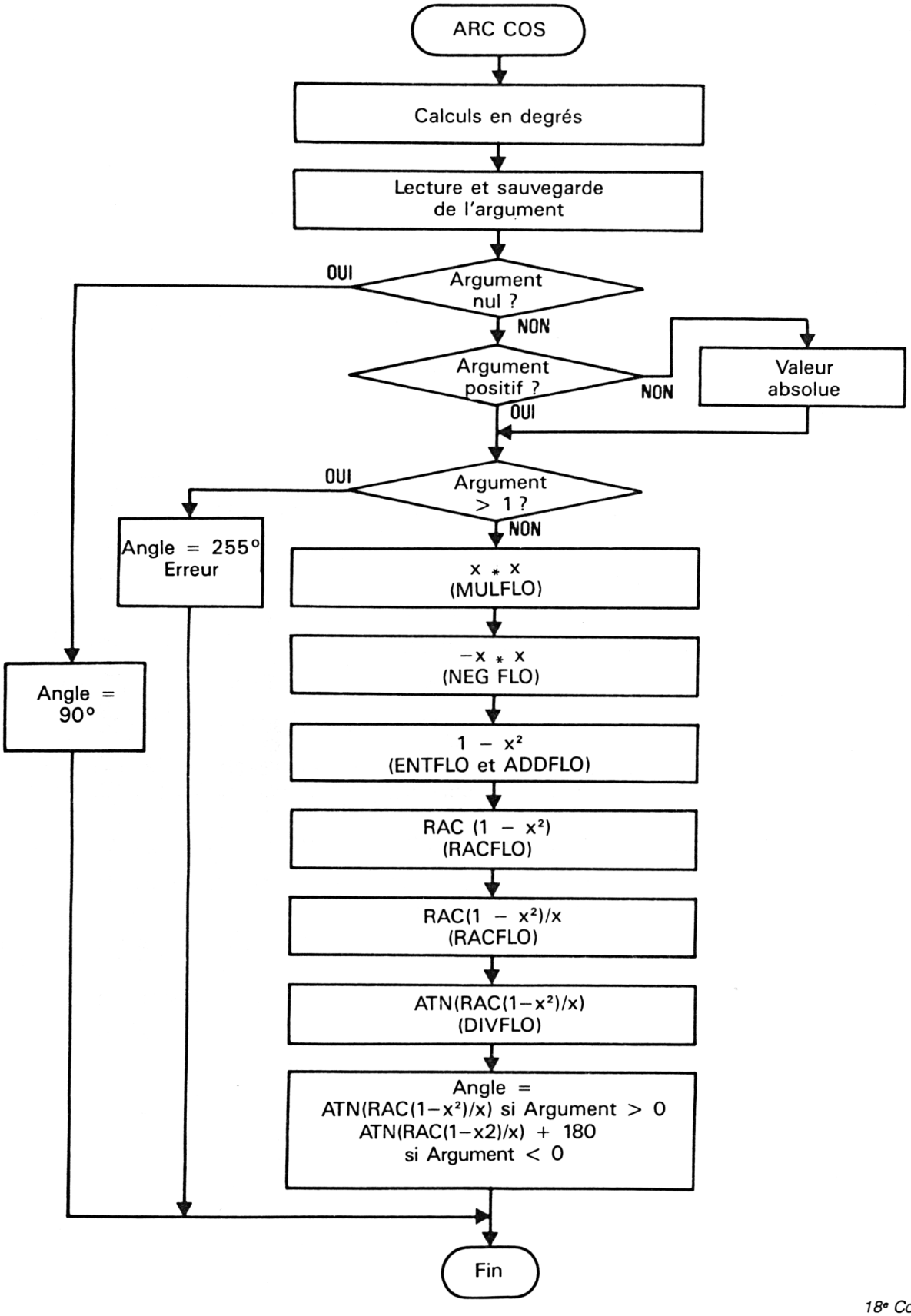
CALL &901F

La fonction |ACOS fait maintenant partie du Basic standard. Voici comment l'utiliser.

```
10 a=.5          'Argument à passer à la fonction ACOS
20 |ACOS,@a      'Appel de la fonction ACOS
30 PRINT a       'Affichage du résultat
```

LE PROGRAMME EN DÉTAIL

La logique du programme obéit à l'ordinogramme suivant : ▶


ARC COS
Calculs en degrés
Lecture et sauvegarde de l'argument
OUI
Argument nul ?
NON
Argument positif ?
NON
Valeur absolue
OUI
OUI
Argument > 1 ?
NON
Angle = 255° Erreur
x * x (MULFLO)
−x * x (NEG FLO)
Angle = 90°
1 − x² (ENTFLO et ADDFLO)
RAC (1 − x²) (RACFLO)
RAC(1 − x²)/x (RACFLO)
ATN(RAC(1−x²)/x) (DIVFLO)
Angle = ATN(RAC(1−x²)/x) si Argument > 0
ATN(RAC(1−x2)/x) + 180 si Argument < 0
Fin

Les premières lignes du programme laissent apparaître de nombreuses déclarations de constantes et variables.

Signalons en particulier :

```
ENTFLO:  EQU  0BD61H  ;Conversion Entier → Flottant
COMFLO:  EQU  0BD8BH  ;Comparaison de deux flottants
SIGFLO:  EQU  0BD91H  ;Signe d'un flottant
NEGFLO:  EQU  0BD8EH  ;Négation d'un flottant
ADDFLO:  EQU  0BD79H  ;Addition de deux flottants
SOUFLO:  EQU  0BD7FH  ;Soustraction de deux flottants
MULFLO:  EQU  0BD82H  ;Multiplication de deux flottants
DIVFLO:  EQU  0BD85H  ;Division de deux flottants
RACFLO:  EQU  0BD9AH  ;Racine d'un flottant
PUIFLO:  EQU  0BD9DH  ;Mise à la puissance d'un flottant
ATNFLO:  EQU  0BDB2H  ;Arc tangente d'un flottant
UNIT:    EQU  0BD94H  ;Unité de calcul des angles
```

Ces adresses sont correctes pour les CPC 664. Si vous possédez un CPC 464 ou un CPC 6128, vous devez les convertir comme suit :

Point d'entrée	CPC 464	CPC 664	CPC 6128
ENTFLO	0BD40H	0BD61H	0BD64H
COMFLO	0BD6AH	0BD8BH	0BD8EH
SIGFLO	0BD70H	0BD91H	0BD94H
NEGFLO	0BD6DH	0BD8EH	0BD91H
ADDFLO	0BD58H	0BD79H	0BD7CH
SOUFLO	0BD5BH	0BD7FH	0BD82H
MULFLO	0BD61H	0BD82H	0BD85H
DIVFLO	0BD64H	0BD85H	0BD88H
RACFLO	0BD79H	0BD9AH	0BD9DH
PUIFLO	0BD7CH	0BD9DH	0BDA0H
ATNFLO	0BD91H	0BDB2H	0BDB5H
UNIT	0BD73H	0BD94H	0BD97H

Remarque :

Pour faciliter l'utilisation de cette RSX, les chargeurs Basic sont donnés dans les trois versions (464, 664 et 6128).

Les zones Z1, Z2 et Z3 de cinq octets sont utilisées pour manipuler les nombres flottants.

La macro LOGEXT permet de déclarer la nouvelle RSX.

Les structures BUF, PTRTAB et TABLE sont typiques des RSX :

— BUF est un buffer utilisé en interne par la RSX.

— PTRTAB contient un pointeur sur la structure TABLE et un débranchement au programme de traitement de la RSX.

— TABLE contient le nom de la RSX. La dernière lettre de ce nom doit être masquée pour que son bit de poids fort soit à un. Enfin, le nom de la dernière RSX doit être suivi d'un octet nul. Dans notre cas, la RSX ACOS étant la seule utilisée, cet octet nul se trouve immédiatement après le nom de la RSX.

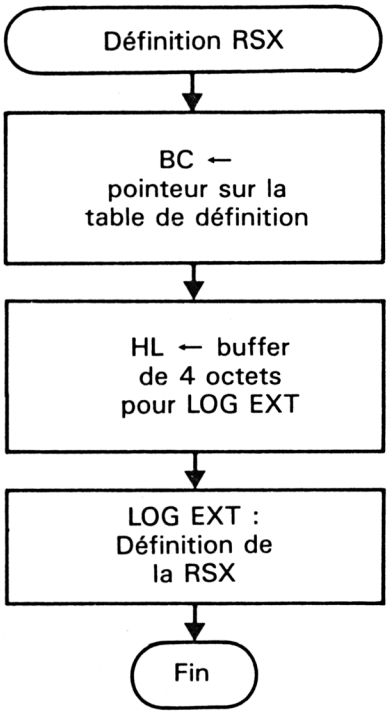
La définition de la RSX doit être effectuée avant sa première utilisation. Le court programme situé à l'étiquette DEFRSX est chargé de cette tâche.

Ce programme fait appel à la macro LOGEXT en lui transmettant :

— l'adresse de la table de définition dans BC,

— l'adresse d'un buffer interne BUF.

La logique de ce programme apparaît dans l'ordinogramme suivant :




```
DEFRSX:   EQU     $
          LD      BC,PTRTAB       ;Ptr table définition
          LD      HL,BUF          ;Buffer pour LOG EXT
          CALL    LOGEXT          ;Définition de la RSX
          RET
```

Lorsque l'utilisateur désire utiliser la fonction **ACOS**, il place l'argument de la fonction dans une variable flottante, et passe l'adresse de cette variable à la RSX **|ACOS.** Le résultat est retourné dans cette même variable.

Par exemple, pour connaître l'arc cosinus de 0.5, tapez :

```
b=0.5:|ACOS,@b: ? b
```

L'ordinateur affichera 60. En effet, acos(0.5) = 60.

Lorsque l'interpréteur rencontre la lettre | (code ASCII 124), il se reporte à la table des RSX à la recherche de la fonction spécifiée. Si la fonction existe dans cette table, elle est exécutée. Dans le cas contraire, un message d'erreur est affiché sur l'écran.

Donc, si vous tapez |ACOS, l'interpréteur recherchera ACOS dans la table RSX. L'adresse de traitement ACOS lui étant affectée, la routine située à l'étiquette ACOS sera exécutée.

La première action effectuée dans cette routine consiste à déclarer que les calculs se feront en degrés :

```
ACOS:   EQU     $
        LD      A,0FFH
        CALL    UNIT            ;Calculs en degrés
```

L'adresse de la variable passée est ensuite récupérée à l'aide du registre IX et stockée dans HL :

```
LD      H,(IX+1)
LD      L,(IX+0)    ;Adresse de la variable
```

Si le nombre passé est nul, le résultat est 90°. Le programme se débranche directement à l'étiquette ANGLE90 :

```
CALL    SIGFLO
OR      A
JP      Z,ANGLE90
```

Si le nombre passé est négatif, son signe est inversé :

```
LD      HL,Z3
CALL    NEGFLO
```

Sa valeur absolue (car toujours positive) est comparée à la constante 1. Si elle est supérieure, la fonction ne peut être exécutée. La valeur 255 est retournée pour signaler une erreur :

```
LD      HL,1
LD      DE,Z2
CALL    ENTFLO
LD      HL,Z3
LD      DE,Z2
CALL    COMFLO
CP      1
JR      Z,ANGLE255
```

Dans le cas où la valeur absolue du nombre passé est inférieure à un, la formule de calcul peut être exécutée :

ATAN(RAC(1 – X^2)/X)

Appelons X la valeur passée. X^2 est calculé à l'aide de la fonction MULFLO. Les deux données multipliées sont égales à X :

```
LD      HL,Z3
LD      DE,Z1
CALL    MULFLO      ; X * X
```

La quantité 1 – X^2 est calculée par modification du signe de X^2 et par addition de la constante 1 :

```
LD      HL,Z3
CALL    NEGFLO      ;– X * X
XOR     A           ;RAZ flag retenue
LD      HL,1
LD      DE,Z2
CALL    ENTFLO      ;1 en flottant
LD      HL,Z2
LD      DE,Z3
CALL    ADDFLO      ;1 – X^2
```

Deux remarques :

— avant de faire la soustraction 1 – X^2, il faut créer le réel 1 en utilisant ENTFLO,

— l'instruction XOR A permet d'effacer l'éventuel signe dans le registre des indicateurs. De cette façon, on est sûr que le nombre converti par ENTFLO sera 1 (et pas – 1),

La racine de 1 – X^2 est obtenue à l'aide du vecteur RACFLO :

```
LD      HL,Z2
CALL    RACFLO      ;RAC(1 – X^2)
```

Cette quantité est ensuite divisée par X à l'aide du vecteur DIVFLO :

```
LD      HL,Z2
LD      DE,Z1
CALL    DIVFLO          ;RAC(1 – X^2)/X
```

Le résultat de ACOS est enfin obtenu en prenant l'arc tangente de la quantité précédente, à l'aide du vecteur ATNFLO :

```
LD      HL,Z2
CALL    ATNFLO          ;ATN(RAC(1 – X^2)/X)
```

Si l'argument est de signe négatif, la valeur 180° est rajoutée au résultat :

```
LD      HL,Z1
CALL    SIGFLO
CP      255
JR      Z,PLUS180       ;Angle + 180
```

Le résultat final est stocké dans la variable passée en entrée à l'aide du sous-programme FLODEHL qui stocke les cinq octets situés à l'adresse pointée par HL à partir de l'adresse pointée par DE :

```
LD      HL,Z2
LD      DE,(SAVHL)
CALL    FLODEHL         ; Résultat
```

Les sous-programmes **ZONE1, ZONE3** et **FLODEHL** fonctionnent selon le même principe :

— le premier permet de stocker les cinq octets pointés par **DE** à partir de la variable **D1**,

— le second permet de stocker les cinq octets pointés par **DE** à partir de la variable **D3**,

— le troisième permet de stocker les cinq octets pointés par **DE** à partir de la variable dont l'adresse est pointée par **HL**.

Chacun de ces programmes utilise la puissante instruction LDIR qui recopie les **BC** octets pointés par **HL** à partir de **DE**. Par exemple, pour **ZONE1** :

```
; HL pointe sur la zone à transférer
LD      DE,Z1       ; Adresse de transfert
LD      BC,5        ; Taille du transfert
LDIR                ; Transfert
```

Si vous préférez utiliser un chargeur Basic, voici le listing et les données de checksum correspondantes :

```
1000 '-------------------------------
1010 ' Chargeur de la RSX ARC COSINUS
1020 '------- -----------------------
1030 ' VERSION CPC 464
1040 '-------------------------------
1050 '
1060 FOR I=&9000 TO &9100
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '-------------------------------
1130 ' Codes op de la RSX ARC COSINUS
1140 '-------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,41,43,4F,D3,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,3E,FF,CD,73,BD,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,E9,90,21,2,90,CD,F2,90
1200 DATA 21,2,90,CD,70,BD,B7,CA,D1,90,FE,FF,20,6,21,C
1210 DATA 90,CD,6D,BD,21,1,0,11,7,90,CD,40,BD,21,C,90
1220 DATA 11,7,90,CD,6A,BD,FE,1,28,73,21,2,90,CD,F2,90
1230 DATA 21,C,90,11,2,90,CD,61,BD,21,C,90,CD,6D,BD,AF
1240 DATA 21,1,0,11,7,90,CD,40,BD,21,7,90,11,C,90,CD
1250 DATA 58,BD,21,7,90,CD,79,BD,21,7,90,11,2,90,CD,64
1260 DATA BD,21,7,90,CD,91,BD,21,2,90,CD,70,BD,FE,FF,28
1270 DATA B,21,7,90,ED,5B,0,90,CD,FB,90,C9,AF,21,B4,0
1280 DATA 11,C,90,CD,40,BD,21,7,90,11,C,90,CD,5B,BD,18
1290 DATA E0,21,5A,0,ED,5B,0,90,CD,40,BD,18,DE,21,FF,0
1300 DATA ED,5B,0,90,CD,40,BD,18,D2,11,2,90,1,5,0,ED
1310 DATA B0,C9,11,C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0
1320 DATA C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
0 D0 10 4D E6 DD 3F B4 CA 62 6A 47 DE 1A 28 3B C9
```

* * *

```
1000 '-------------------------------
1010 ' Chargeur de la RSX ARC COSINUS
1020 '-------------------------------
1030 ' VERSION CPC664
1040 '-------------------------------
1050 '
1060 FOR I=&9000 TO &9100
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '-------------------------------
1130 ' Codes op de la RSX ARC COSINUS
1140 '-------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,41,43,4F,D3,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,3E,FF,CD,94,BD,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,E9,90,21,2,90,CD,F2,90
1200 DATA 21,2,90,CD,91,BD,B7,CA,D1,90,FE,FF,20,6,21,C
1210 DATA 90,CD,8E,BD,21,1,0,11,7,90,CD,61,BD,21,C,90
1220 DATA 11,7,90,CD,8B,BD,FE,1,28,73,21,2,90,CD,F2,90
1230 DATA 21,C,90,11,2,90,CD,82,BD,21,C,90,CD,8E,BD,AF
1240 DATA 21,1,0,11,7,90,CD,61,BD,21,7,90,11,C,90,CD
1250 DATA 79,BD,21,7,90,CD,9A,BD,21,7,90,11,2,90,CD,85
1260 DATA BD,21,7,90,CD,B2,BD,21,2,90,CD,91,BD,FE,FF,28
1270 DATA B,21,7,90,ED,5B,0,90,CD,FB,90,C9,AF,21,B4,0
1280 DATA 11,C,90,CD,61,BD,21,7,90,11,C,90,CD,7F,BD,18
1290 DATA E0,21,5A,0,ED,5B,0,90,CD,61,BD,18,DE,21,FF,0
1300 DATA ED,5B,0,90,CD,61,BD,18,D2,11,2,90,1,5,0,ED
1310 DATA B0,C9,11,C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0
1320 DATA C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
0 D0 31 4D 8 20 60 F6 EB C5 AC 47 24 3B 49 3B C9
```

```
1000 '--------------------------------
1010 ' Chargeur de la RSX ARC COSINUS
1020 '--------------------------------
1030 ' VERSION CPC 6128
1040 '--------------------------------
1050 '
1060 FOR I=&9000 TO &9100
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '--------------------------------
1130 ' Codes op de la RSX ARC COSINUS
1140 '--------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,41,43,4F,D3,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,3E,FF,CD,97,BD,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,E9,90,21,2,90,CD,F2,90
1200 DATA 21,2,90,CD,94,BD,B7,CA,D1,90,FE,FF,20,6,21,C
1210 DATA 90,CD,91,BD,21,1,0,11,7,90,CD,64,BD,21,C,90
1220 DATA 11,7,90,CD,8E,BD,FE,1,28,73,21,2,90,CD,F2,90
1230 DATA 21,C,90,11,2,90,CD,85,BD,21,C,90,CD,91,BD,AF
1240 DATA 21,1,0,11,7,90,CD,64,BD,21,7,90,11,C,90,CD
1250 DATA 7C,BD,21,7,90,CD,9D,BD,21,7,90,11,2,90,CD,88
1260 DATA BD,21,7,90,CD,B5,BD,21,2,90,CD,94,BD,FE,FF,28
1270 DATA B,21,7,90,ED,5B,0,90,CD,FB,90,C9,AF,21,B4,0
1280 DATA 11,C,90,CD,64,BD,21,7,90,11,C,90,CD,82,BD,18
1290 DATA E0,21,5A,0,ED,5B,0,90,CD,64,BD,18,DE,21,FF,0
1300 DATA ED,5B,0,90,CD,64,BD,18,D2,11,2,90,1,5,0,ED
1310 DATA B0,C9,11,C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0
1320 DATA C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
0 D0 34 4D B 26 63 FC EE CE B2 47
 2A 3E 4C 3B C9
```