

9/4.5.4

Analyse combinatoire

INTRODUCTION

Parfois, il peut être utile de connaître le nombre de combinaisons possibles de P objets parmi N . Par exemple dans les jeux pour lesquels le hasard entre en ligne de compte. L'exemple type est la course de chevaux. Supposons 17 chevaux au départ d'une course. Comment calculer le nombre de tiercés possibles dans l'ordre et dans le désordre ?

La notion de tiercé dans l'ordre correspond à celle d'**arrangement**, et la notion de tiercé dans le désordre à celle de **combinaison**.

Le nombre de tiercés possibles dans l'ordre pour 17 chevaux est égal à :

$$A_{17}^3$$

Un arrangement est calculé à l'aide de factorielles selon la formule suivante :

$$A_n^p = \frac{n!}{(n-p)!}$$

Le nombre de tiercés possibles dans le désordre pour 17 chevaux est égal à :

$$C_{17}^3$$

Une combinaison est calculée à l'aide de factorielles selon la formule suivante :

$$C_n^p = \frac{n!}{p!(n-p)!}$$

Les RSX présentées dans ce complément permettent de calculer des factorielles (RSX | **FACT**) en entier et en flottant, des arrangements (RSX | **ANP**) en flottant et des combinaisons (RSX | **CNP**) en flottant.

CALCUL DE FACTORIELLES

Point de départ :

- Comme chacun le sait,

$$N ! = N \times (N-1) \times \dots \times 1$$

Le calcul à effectuer est donc simple. Le nombre N étant toujours entier, il semble logique d'effectuer le calcul en entiers.

- En assembleur Z80, les multiplications entières n'existent pas. Pour calculer la quantité

$A \times B$

il faut en fait calculer la quantité

$$A + A + \dots + A \text{ (B fois)}$$

- Nous devons transmettre un paramètre à la RSX : le nombre dont nous désirons connaître la factorielle. Soit N ce nombre.

En sortie, nous désirons récupérer la quantité N !

Le format de la RSX est donc :

| FACT, @N%

où N% est une variable entière dans laquelle est stocké le nombre N.

Ces notions étant acquises, écrivons la RSX |FACT.

La RSX |FACT en détail

Le listing de la RSX est le suivant :

```

1          ORG  9000H
2          LOAD 9000H
3          ;-----
4          ; RSX FACT
5          ; Format : IFACT,@N%
6          ; Entree : N%=Variable entiere
7          ; Sortie : N%=Factorielle N%
8          ;-----
9          ;
10         ;
11         ;-----
12         ; Declaration des constantes
13         ; et des variables du programme
14         ;-----
15         ;
16         LOGEXT:  EQU  0BCD1H          ;KL LOG EXT
17         BUF:     DS   4                ;ZONE RAM POUR LOG EXT
18 9004 0990      PTRTAB:  DW  TABLE    ;Pointeur TABLE
19 9006 C31A90          JP  FACT        ;Calcul de la factorielle
20 9009 464143      TABLE:  DB  "FAC"
21 900C D4          DB  "T"+80H
22 900D 00          DB  0                ;Fin de table
23         ADRESSE:  DS   2                ;Adresse variable
24         ;
25         ;-----
26         ; Definition de la RSX
27         ;-----
28         ;
29         DEFRSX:  EQU  *                ;Point d'entree
30 9010 010490      LD  BC,PRTAB        ;Ptr table definition
31 9013 210090      LD  HL,BUF         ;Buffer pour LOG EXT

```

```

32 9016 CDD1BC          CALL LOGEXT          ;Definition de la RSX
33 9019 C9             RET
34                    ;
35                    ;-----
36                    ; Traitement de FACT
37                    ;-----
38                    ;
39                    FACT:      EQU  $          ;Point d'entree
40                    ;
41                    ;-----
42                    ; Lecture de la donnee passee
43                    ;-----
44                    ;
45 901A DD6601          LD   H, (IX+1)
46 901D DD6E00          LD   L, (IX+0)
47 9020 220E90          LD   (ADRESSE),HL
48 9023 7E             LD   A, (HL)          ;Donnee de base
49                    ;
50                    ;-----
51                    ; Calcul de la factorielle
52                    ;-----
53                    ;
54 9024 2600          LD   H,0
55 9026 6F             LD   L,A          ;Nombre de base
56                    BIS2:      EQU  $
57 9027 54             LD   D,H
58 9028 5D             LD   E,L
59 9029 3D             DEC  A
60 902A FE01          CP   1

```

```

61 902C 2807      JR   Z,FIN           ;Fin du calcul
62 902E 47       LD   B,A
63 902F 05       DEC  B
64              BIS:   EQU  $
65 9030 19       ADD  HL,DE
66 9031 10FD     DJNZ BIS           ;Addition
67 9033 18F2     JR   BIS2          ;Multiplication
68              FIN:   EQU  $
69 9035 ED5B0E90 LD   DE,(ADRESSE)
70 9039 EB       EX   DE,HL
71 903A 73       LD   (HL),E       ;Sauvegarde LSB
72 903B 23       INC  HL
73 903C 72       LD   (HL),D       ;Sauvegarde MSB
74              ;
75 903D C9       RET
76              END

```

Entre les lignes 16 et 22, nous retrouvons les données caractéristiques des RSX. En ligne 23 est définie la variable **ADRESSE** dans laquelle sera stockée l'adresse de la variable qui contient le paramètre.

La routine **DEFRSX** (lignes 29 à 33) est désormais classique. Elle initialise la RSX **FACT**.

Le traitement associé à la RSX se trouve entre les lignes 39 et 75. La première action effectuée consiste à récupérer :

– l'adresse du paramètre et à la stocker dans la variable **ADRESSE** lignes 45 à 47 ;

```

LD   H, (IX + 1)
LD   L, (IX + 0)
LD   (ADRESSE), HL

```

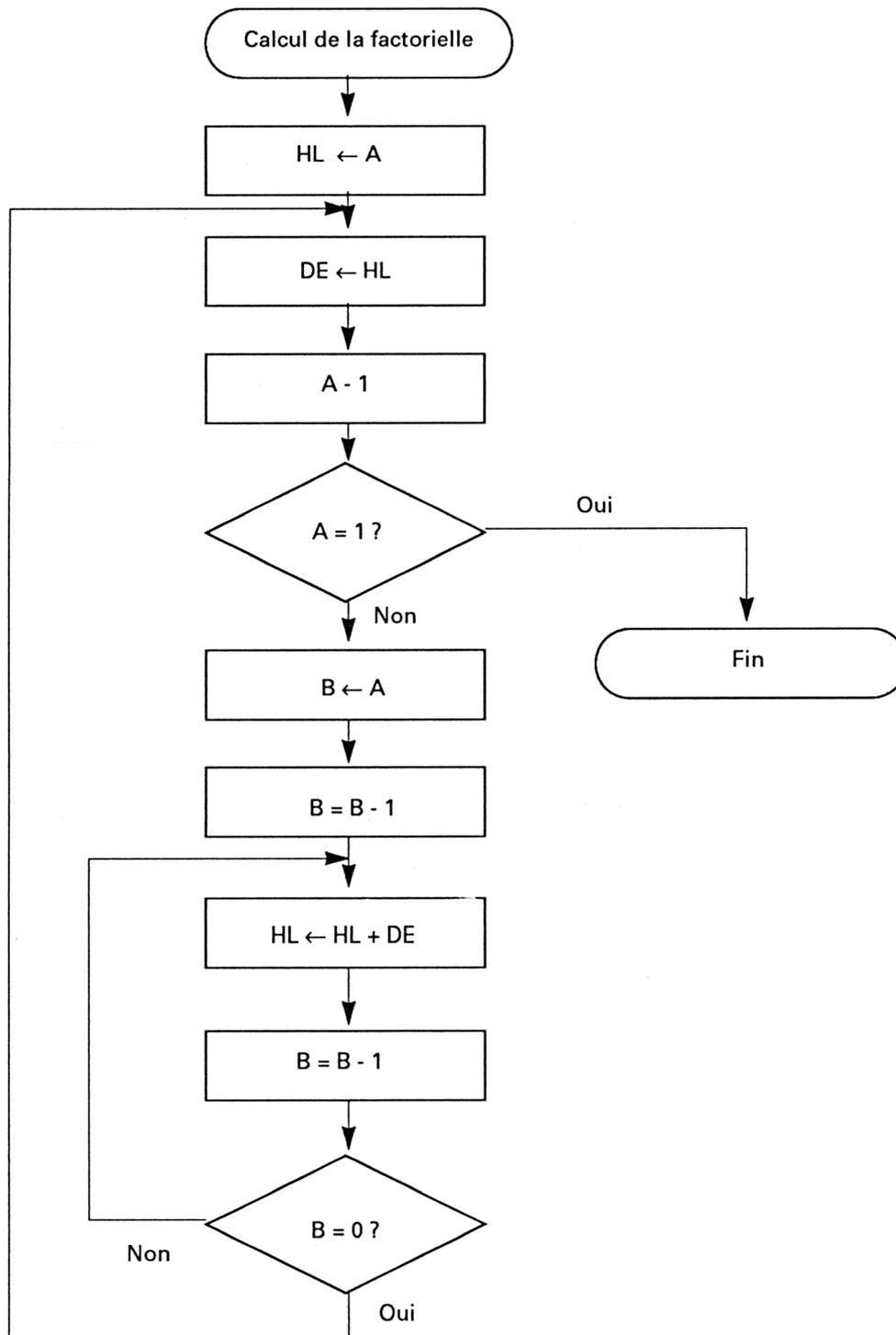
– le paramètre et à le stocker dans le registre **A** ligne 48 :

```

LDA, (HL)

```

Ces préliminaires effectués, deux boucles imbriquées effectuent les multiples additions nécessaires au calcul de la factorielle selon la logique de l'ordinogramme suivant :



La paire de registres **HL** contient le résultat intermédiaire et final. Elle est initialisée à la valeur contenue dans le registre **A** lignes 54 et 55.

La paire de registres **DE** contient le prochain facteur multiplicatif.

Le registre **B** contient le nombre d'additions à réaliser pour effectuer la multiplication courante. Ce registre est décrémenté avec l'instruction **DJNZ** qui effectue une boucle tant que **B** est différent de zéro (lignes 64 à 66).

La seconde boucle décrémente le facteur multiplicatif et arrête le processus lorsqu'il est égal à 1 (lignes 67 et 59 à 61).

Lorsque toutes les multiplications sont calculées, le résultat est transféré à l'adresse contenue dans la variable **ADRESSE**, c'est-à-dire dans le paramètre d'entrée (lignes 68 à 75).

Voici le chargeur de la RSX **FACT** :

```

1000 '-----
1010 ' Chargeur de la RSX FACT
1020 '-----
1030 FOR i=&9000 TO &903D
1040   READ a#
1050   a=VAL("&"+a#)
1060   POKE i,a
1070 NEXT i
1080 CALL &9010
1090 '
1100 '-----
1110 ' Codes Op de la RSX FACT
1120 '-----
1130 DATA 0,0,0,0,9,90,C3,1A,90,46,41,43,D4,0,0,0
1140 DATA 1,4,90,21,0,90,CD,D1,BC,C9,DD,66,1,DD,6E,0
1150 DATA 22,E,90,7E,26,0,6F,54,5D,3D,FE,1,28,7,47,5
1160 DATA 19,10,FD,18,F2,ED,5B,E,90,EB,73,23,72,C9,0,0

```

et les données de checksum correspondantes.

```
A7 FE 3F D8
```

Exécutez ce programme. La RSX est désormais utilisable. Essayez par exemple en mode direct :

```
a%=4: !FACT, @a%:PRINT a%
```

L'ordinateur affiche la valeur 24 qui est bien le résultat attendu.

Essayez d'autres valeurs. Par exemple :

```
a%=7: !FACT, @a%:PRINT a%
```

L'ordinateur affiche 5040. C'est bien la valeur attendue.

Essayez maintenant :

```
a%=8: !FACT, @a%:PRINT a%
```

L'ordinateur affiche -25216. Il y a dépassement de capacité !

La RSX donne donc de bons résultats pour des paramètres entiers inférieurs à 8. Pour pallier ce problème, nous allons réécrire la RSX en format réel.

La forme générale de la RSX est la même. Seuls les calculs sont plus complexes, car ils portent sur des nombres réels. En voici le listing en assembleur :

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; RSX FACT
5          ; Format : IFACT,@N
6          ; Entree : N=Variable reelle
7          ; Sortie : N=Factorielle N
8          ;-----
9          ;
10         ;
11         ;-----
12         ; Declaration des constantes
13         ; et des variables du programme
14         ;-----
15         ;
16         FLOENT: EQU 0BD67H          ;Flot->Entier
17         ADDFLO: EQU 0BD79H          ;Addition flot
18         LOGEXT: EQU 0BCD1H          ;KL LOG EXT
19         BUF: DS 4                   ;ZONE RAM POUR LOG EXT
20 9004 0990 PTRTAB: DW TABLE         ;Pointeur TABLE
21 9006 C32490 JP FACT                 ;Calcul de la factorielle
22 9009 464143 TABLE: DB "FAC"
23 900C D4 DB "T"+80H
24 900D 00 DB 0                       ;Fin de table
25         SAVHL: DS 2                 ;Adresse du parametre
26         Z1: DS 5                    ;Zone reel 1
27         Z2: DS 5                    ;Zone reel 2
28         ;
29         ;-----
30         ; Definition de la RSX
31         ;-----

```

```

32          ;
33  DEFRSX:  EQU  $          ;Point d'entree
34  901A 010490      LD  BC, PTRTAB      ;Ptr table definition
35  901D 210090      LD  HL, BUF        ;Buffer pour LOG EXT
36  9020 CDD1BC      CALL LOGEXT       ;Definition de la RSX
37  9023 C9          RET
38          ;
39          ;-----
40          ; Traitement de FACT
41          ;-----
42          ;
43  FACT:      EQU  $          ;Point d'entree
44          ;
45          ;-----
46          ; Lecture de la donnee passee
47          ;-----
48          ;
49  9024 DD6601      LD  H, (IX+1)
50  9027 DD6E00      LD  L, (IX+0)
51  902A 220E90      LD  (SAVHL), HL
52  902D CD6590      CALL ZONE1        ;Memorisation
53  9030 211090      LD  HL, Z1
54  9033 CD67BD      CALL FLOENT       ;Conversion entiere
55  9036 7D          LD  A, L          ;Sauvegarde
56          ;
57          ;-----
58          ; Calcul de la factorielle
59          ;-----
60          ;

```

```

61          BIS2:    EQU  $
62 9037 211090      LD  HL,Z1
63 903A 111590      LD  DE,Z2
64 903D F5          PUSH AF
65 903E CD6E90      CALL FLODEHL      ;Z2=Z1
66 9041 F1          POP  AF
67 9042 3D          DEC  A
68 9043 FE01        CP   1
69 9045 2813        JR   Z,FIN      ;Fin du calcul
70 9047 47          LD  B,A
71 9048 05          DEC  B
72          BIS:    EQU  $
73 9049 F5          PUSH AF
74 904A C5          PUSH BC
75 904B 211090      LD  HL,Z1
76 904E 111590      LD  DE,Z2
77 9051 CD79BD      CALL ADDFLO
78 9054 C1          POP  BC
79 9055 F1          POP  AF
80 9056 10F1        DJNZ BIS      ;Addition
81 9058 18DD        JR   BIS2     ;Multiplication
82          FIN:    EQU  $
83 905A 211090      LD  HL,Z1
84 905D ED5B0E90    LD  DE,(SAVHL)
85 9061 CD6E90      CALL FLODEHL     ;Resultat
86          ;
87 9064 C9          RET
88          ;
89          ;-----
90          ; Zone des sous-programmes
91          ;-----

```

```

92          ;
93          ;-----
94          ; Transfert des BC octets pointes
95          ; par HL dans le buffer Z1
96          ;-----
97          ;
98          ZONE1:      EQU  $
99 9065 111090          LD   DE,Z1
100 9068 010500        LD   BC,S
101 906B EDB0          LDIR
102 906D C9            RET
103          ;
104          ;-----
105          ; Transfert flottant de (HL)
106          ; dans (DE)
107          ;-----
108          ;
109          FLODEHL:   EQU  $
110 906E 010500        LD   BC,S
111 9071 EDB0          LDIR
112 9073 C9            RET
113          END

```

Les points d'entrée des routines FLOENT et ADDFLO qui apparaissent dans le listing sont valables sur les CPC 664. Pour faciliter la saisie, nous vous proposons le chargeur Basic dans les trois versions de CPC.

Version CPC 464 :

```
1000 '-----
1010 ' Chargeur de la RSX FACT
1020 ' Version CPC 464
1030 '-----
1040 FOR i=&9000 TO &9073
1050   READ a$
1060   a=VAL("&" + a$)
1070   POKE i,a
1080 NEXT i
1090 CALL &901A
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,9,90,C3,24,90,46,41,43,D4,0,0,0
1150 DATA 0,0,0,0,0,0,0,0,0,0,1,4,90,21,0,90
1160 DATA CD,D1,BC,C9,DD,66,1,DD,6E,0,22,E,90,CD,65,90
1170 DATA 21,10,90,CD,46,BD,7D,21,10,90,11,15,90,F5,CD,6E
1180 DATA 90,F1,3D,FE,1,28,13,47,5,F5,C5,21,10,90,11,15
1190 DATA 90,CD,58,BD,C1,F1,10,F1,18,DD,21,10,90,ED,5B,E
1200 DATA 90,CD,6E,90,C9,11,10,90,1,5,0,ED,B0,C9,1,5
1210 DATA 0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0,0,0
```

B1 47 3C BB EA 39 4D

Version CPC 664 :

```
1000 '-----
1010 ' Chargeur de la RSX FACT
1020 ' Version CPC 664
1030 '-----
1040 FOR i=&9000 TO &9073
1050   READ a$
1060   a=VAL("&" + a$)
1070   POKE i,a
1080 NEXT i
1090 CALL &901A
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,9,90,C3,24,90,46,41,43,D4,0,0,0
1150 DATA 0,0,0,0,0,0,0,0,0,0,1,4,90,21,0,90
1160 DATA CD,D1,BC,C9,DD,66,1,DD,6E,0,22,E,90,CD,65,90
1170 DATA 21,10,90,CD,67,BD,7D,21,10,90,11,15,90,F5,CD,6E
1180 DATA 90,F1,3D,FE,1,28,13,47,5,F5,C5,21,10,90,11,15
1190 DATA 90,CD,79,BD,C1,F1,10,F1,18,DD,21,10,90,ED,5B,E
1200 DATA 90,CD,6E,90,C9,11,10,90,1,5,0,ED,B0,C9,1,5
1210 DATA 0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

B1 47 3C DC EA 5A 4D

Version CPC 6128 :

```

1000 '-----
1010 ' Chargeur de la RSX FACT
1020 ' Version CPC 6128
1030 '-----
1040 FOR i=&9000 TO &9073
1050   READ a$
1060   a=VAL("&" + a$)
1070   POKE i,a
1080 NEXT i
1090 CALL &901A
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,9,90,C3,24,90,46,41,43,D4,0,0,0
1150 DATA 0,0,0,0,0,0,0,0,0,0,1,4,90,21,0,90
1160 DATA CD,D1,BC,C9,DD,66,1,DD,6E,0,22,E,90,CD,65,90
1170 DATA 21,10,90,CD,6A,BD,7D,21,10,90,11,15,90,F5,CD,6E
1180 DATA 90,F1,3D,FE,1,28,13,47,5,F5,C5,21,10,90,11,15
1190 DATA 90,CD,7C,BD,C1,F1,10,F1,18,DD,21,10,90,ED,5B,E
1200 DATA 90,CD,6E,90,C9,11,10,90,1,5,0,ED,B0,C9,1,5
1210 DATA 0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0,0,0,0

```

B1 47 3C DF EA 5D 4D

La RSX : FACT réelle en détail

La zone de déclaration de la RSX réelle est assez similaire à celle de la RSX entière. Remarquez simplement :

- les routines de manipulation de réels :
 - FLOENT : conversion d'un flottant en entier (ligne 16) ;
 - ADDFLO : addition de deux flottants (ligne 17) ;
- les variables réelles Z1 et Z2 déclarées sur 5 octets lignes 26 et 27.

La routine de définition de la RSX est la même que celle de la version entière.

La routine de traitement de la RSX est un peu plus complexe. Remarquez la sauvegarde du paramètre passé à la RSX lignes 49 à 55 :

```

LD   H, (IX+1)
LD   L, (IX+0)
LD   (SAVHL), HL   →   adresse du paramètre ;
CALLZONE1           →   sauvegarde du paramètre dans Z1 ;

```

```
LD  HL, Z1
CALLFLOENT    →  conversion entière de Z1 ;
LD  A, L      →  et sauvegarde dans le registre A.
```

Le calcul de la factorielle se fait toujours à l'aide de deux boucles imbriquées. Les additions se font avec la routine **ADDFLO** (ligne 77). En fin de calcul, le résultat est stocké dans le paramètre passé en entrée à l'aide du sous-programme **FLODEHL** (ligne 85).

Le programme se termine par deux sous-programmes de manipulation de nombres réels.

- **ZONE1** transfère le nombre réel (5 octets) pointé par **HL** dans la variable **Z1** ;
- **FLODEHL** transfère le nombre réel (5 octets) pointé par **HL** dans **DE**.

ARRANGEMENTS ET COMBINAISONS

Nous allons terminer notre exposé avec un programme plus important organisé en trois RSX :

```
! FACT → factorielle ;
! ANP  → arrangement N, P ;
! CNP  → combinaison N, P.
```

Le listing assembleur du programme est le suivant :

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          ; RSX FACT, ANP et CNP
5          ; Format : !FACT,@N
6          ;          !ANP,@N,@P
7          ;          !CNP,@N,@P
8          ; Entree : N=Variable reelle
9          ;          P=Variable reelle
10         ; Sortie : N=Factorielle N
11         ;          N=Arrangement N,P
12         ;          N=Combinaison N,P
13         ;-----
14         ;
15         ;
16         ;-----
17         ; Declaration des constantes
18         ; et des variables du programme
19         ;-----
20         ;
21         FLOENT: EQU 0BD67H          ;Flot->Entier
22         ADDFLO: EQU 0BD79H          ;Addition flot
23         NEGFLO: EQU 0BD8EH          ;Negation flot
24         MULFLO: EQU 0BD82H          ;Multipl flot
25         DIVFLO: EQU 0BD85H          ;Division flot
26         LOGEXT: EQU 0BCD1H          ;KL LOG EXT
27         ;
28         BUF: DS 4                   ;ZONE RAM POUR LOG EXT
29 9004 0F90 PTRTAB: DW TABLE          ;Pointeur TABLE
30 9006 C34690 JP FACT                  ;Calcul de la factorielle
31 9009 C38790 JP ANP                   ;Calcul de l'arrangement

```

```

32 900C 03D490      JP    CNP                ;Calcul de la combinaison
33 900F 464143      TABLE: DB    "FAC"
34 9012 D4          DB    "T"+80H
35 9013 414E        DB    "AN"
36 9015 D0          DB    "P"+80H
37 9016 434E        DB    "CN"
38 9018 D0          DB    "P"+80H
39 9019 00          DB    0                ;Fin de table
40                  ;
41                  SAVHL: DS    2                ;Adresse du parametre
42                  SAVHL2: DS   2                ;Adresse du parametre
43                  Z1:      DS    5                ;Zone reel 1
44                  Z2:      DS    5                ;Zone reel 2
45                  Z3:      DS    5                ;Zone reel 3
46                  Z4:      DS    5                ;Zone reel 4
47                  Z5:      DS    5                ;Zone reel 5
48                  Z6:      DS    5                ;Zone reel 6
49                  ;
50                  ;-----
51                  ; Definition des RSX
52                  ;-----
53                  ;
54                  DEFRSX: EQU  $                ;Point d'entree
55 903C 010490      LD    BC,PTRTAB        ;Ptr table definition
56 903F 210090      LD    HL,BUF           ;Buffer pour LOG EXT
57 9042 CDD1BC      CALL LOGEXT           ;Definition de la RSX
58 9045 C9          RET
59                  ;
60                  ;

```

```

61          ;-----
62          ; Traitement de FACT
63          ;-----
64          ;
65          FACT:      EQU $           ;Point d'entree
66          ;
67          ;-----
68          ; Lecture des donnees passees
69          ;-----
70          ;
71 9046 DD6601          LD   H,(IX+1)
72 9049 DD6E00          LD   L,(IX+0)
73          EINT:      EQU $           ;Pt d'entree interne
74 904C 221A90          LD   (SAVHL),HL
75 904F CD3991          CALL ZONE1      ;Memorisation
76 9052 211E90          LD   HL,Z1
77 9055 CD67BD          CALL FLOENT      ;Conversion entiere
78 9058 7D             LD   A,L           ;Sauvegarde
79          ;
80          ;-----
81          ; Calcul de la factorielle
82          ;-----
83          ;
84          BIS2:      EQU $
85 9059 211E90          LD   HL,Z1
86 905C 112390          LD   DE,Z2
87 905F F5             PUSH AF
88 9060 CD5491          CALL FLODEHL      ;Z2=Z1
89 9063 F1             POP  AF
90 9064 3D             DEC  A
91 9065 FE01          CP   1
    
```

```

92 9067 2813      JR   Z,FIN           ;Fin du calcul
93 9069 47        LD   B,A
94 906A 05        DEC  B
95                BIS:   EQU  $
96 906B F5        PUSH AF
97 906C C5        PUSH BC
98 906D 211E90    LD   HL,Z1
99 9070 112390    LD   DE,Z2
100 9073 CD79BD   CALL ADDFLO
101 9076 C1       POP  BC
102 9077 F1       POP  AF
103 9078 10F1     DJNZ BIS           ;Addition
104 907A 18DD     JR   BIS2           ;Multiplication
105                FIN:   EQU  $
106 907C 211E90    LD   HL,Z1
107 907F ED5B1A90 LD   DE,(SAVHL)
108 9083 CD5491   CALL FLODEHL       ;Resultat
109                ;
110 9086 C9        RET
111                ;
112                ;-----
113                ; Traitement de ANP
114                ;-----
115                ;
116                ANP:   EQU  $           ;Point d'entree
117                ;
118                ;-----
119                ; Lecture des donnees passees
120                ;-----

```

```

121          ;
122 9087 DD6601          LD   H, (IX+1)
123 908A DD6E00          LD   L, (IX+0)
124 908D CD4B91          CALL ZONE5          ;Memo 2eme argument
125 9090 DD6603          LD   H, (IX+3)
126 9093 DD6E02          LD   L, (IX+2)
127 9096 221090          LD   (SAVHL2),HL
128 9099 CD4291          CALL ZONE4          ;Memo 1er argument
129          ;
130          ;- - - - -
131          ; Calcul de ANP
132          ;- - - - -
133          ;
134 909C 212D90          LD   HL,Z4
135 909F 113790          LD   DE,Z6
136 90A2 CD5491          CALL FLODEHL          ;Sauv pr calc N!
137 90A5 213790          LD   HL,Z6
138 90A8 CD4C90          CALL EINT          ;N!
139          ;
140 90AB 213290          LD   HL,Z5
141 90AE CD8E8D          CALL NEGFLD          ;-P
142 90B1 212D90          LD   HL,Z4
143 90B4 113290          LD   DE,Z5
144 90B7 CD798D          CALL ADDFLD          ;N-P
145 90BA 212D90          LD   HL,Z4
146 90BD CD4C90          CALL EINT          ; (N-P) !
147          ;
148 90C0 213790          LD   HL,Z6
149 90C3 112D90          LD   DE,Z4
150 90C6 CD858D          CALL DIVFLD
151 90C9 213790          LD   HL,Z6

```

```

152 90CC ED5B1C90          LD   DE, (SAVHL2)
153 90D0 CD5491           CALL FLODEHL           ;Resultat
154 90D3 C9              RET
155                       ;
156                       ;-----
157                       ; Traitement de CNP
158                       ;-----
159                       ;
160 CNP:                   EQU $           ;Point d'entree
161                       ;
162                       ;-----
163                       ; Lecture des donnees passees
164                       ;-----
165                       ;
166 90D4 DD6601           LD   H, (IX+1)
167 90D7 DD6E00           LD   L, (IX+0)
168 90DA CD4B91           CALL ZONE5            ;Memo 2eme argument
169 90DD DD6603           LD   H, (IX+3)
170 90E0 DD6E02           LD   L, (IX+2)
171 90E3 221C90          LD   (SAVHL2),HL
172 90E6 CD4291           CALL ZONE4            ;Memo 1er argument
173                       ;
174                       ;-----
175                       ; Calcul de CNP
176                       ;-----
177                       ;
178 90E9 212D90          LD   HL,Z4
179 90EC 113790          LD   DE,Z6
180 90EF CD5491           CALL FLODEHL           ;Sauv pr calc N!

```

```

181 90F2 213790      LD   HL,Z6
182 90F5 CD4C90      CALL EINT           ;N!
183                  ;
184 90F8 213290      LD   HL,Z5
185 90FB 112890      LD   DE,Z3
186 90FE CD5491      CALL FLODEHL       ;Sauv pr calc P!
187 9101 212890      LD   HL,Z3
188 9104 CD4C90      CALL EINT           ;P!
189                  ;
190 9107 213290      LD   HL,Z5
191 910A CDBEBD      CALL NEGFLD        ;-P
192 910D 212D90      LD   HL,Z4
193 9110 113290      LD   DE,Z5
194 9113 CD79BD      CALL ADDFLD        ;N-P
195 9116 212D90      LD   HL,Z4
196 9119 CD4C90      CALL EINT           ; (N-P)!
197                  ;
198 911C 212890      LD   HL,Z3
199 911F 112D90      LD   DE,Z4
200 9122 CD82BD      CALL MULFLD        ;P! * (N-P)!
201 9125 213790      LD   HL,Z6
202 9128 112890      LD   DE,Z3
203 912B CD85BD      CALL DIVFLD        ;N! / (P! * (N-P)!)
204 912E 213790      LD   HL,Z6
205 9131 ED5B1C90     LD   DE,(SAVHL2)
206 9135 CD5491      CALL FLODEHL       ;Resultat
207 9138 C9          RET
208                  ;
209                  ;-----
210                  ; Zone des sous-programmes
211                  ;-----

```

```
212          ;
213          ;-----
214          ; Transfert des BC octets pointes
215          ; par HL dans Z1, Z4 ou Z5
216          ;-----
217          ;
218          ZONE1:      EQU  $
219 9139 111E90          LD   DE,Z1
220 913C 010500          LD   BC,5
221 913F EDB0           LDIR
222 9141 C9             RET
223          ;
224          ZONE4:      EQU  $
225 9142 112D90          LD   DE,Z4
226 9145 010500          LD   BC,5
227 9148 EDB0           LDIR
228 914A C9             RET
229          ;
230          ZONE5:      EQU  $
231 914B 113290          LD   DE,Z5
232 914E 010500          LD   BC,5
233 9151 EDB0           LDIR
234 9153 C9             RET
235          ;
241          FLODEHL:   EQU  $
242 9154 010500          LD   BC,5
243 9157 EDB0           LDIR
244 9159 C9             RET
245          END
```

Il est écrit pour fonctionner sur CPC 664. Le chargeur Basic correspondant est proposé dans les trois versions de CPC.

Version CPC 464 :

```

1000 '-----
1010 ' Chargeur des RSX FACT, ANP et CNP
1020 ' Version CPC 464
1030 '-----
1040 FOR i=&9000 TO &9159
1050   READ a#
1060   a=VAL("&" + a#)
1070   POKE i,a
1080 NEXT i
1090 CALL &903C
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,F,90,C3,46,90,C3,87,90,C3,D4,90,46
1150 DATA 41,43,D4,41,4E,D0,43,4E,D0,0,0,0,0,0,0,0
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,4,90,21
1180 DATA 0,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22,1A,90,CD
1190 DATA 39,91,21,1E,90,CD,46,BD,7D,21,1E,90,11,23,90,F5
1200 DATA CD,54,91,F1,3D,FE,1,28,13,47,5,F5,C5,21,1E,90
1210 DATA 11,23,90,CD,58,BD,C1,F1,10,F1,18,DD,21,1E,90,ED
1220 DATA 5B,1A,90,CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91
1230 DATA DD,66,3,DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11
1240 DATA 37,90,CD,54,91,21,37,90,CD,4C,90,21,32,90,CD,6D
1250 DATA BD,21,2D,90,11,32,90,CD,58,BD,21,2D,90,CD,4C,90
1260 DATA 21,37,90,11,2D,90,CD,64,BD,21,37,90,ED,5B,1C,90
1270 DATA CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91,DD,66,3
1280 DATA DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11,37,90,CD
1290 DATA 54,91,21,37,90,CD,4C,90,21,32,90,11,28,90,CD,54
1300 DATA 91,21,28,90,CD,4C,90,21,32,90,CD,6D,BD,21,2D,90
1310 DATA 11,32,90,CD,58,BD,21,2D,90,CD,4C,90,21,28,90,11
1320 DATA 2D,90,CD,61,BD,21,37,90,11,28,90,CD,64,BD,21,37
1330 DATA 90,ED,5B,1C,90,CD,54,91,C9,11,1E,90,1,5,0,ED
1340 DATA B0,C9,11,2D,90,1,5,0,ED,B0,C9,11,32,90,1,5
1350 DATA 0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0

```

85 1C 0 B6 E2 74 F5 12 BF F5 2E DD 86 1 44 49 D1 2C A5 B7 91 D6

Version CPC 664 :

```

1000 '-----
1010 ' Chargeur des RSX FACT, ANP et CNP
1020 ' Version CPC 664
1030 '-----
1040 FOR i=&9000 TO &9159
1050   READ a#
1060   a=VAL("&"+a#)
1070   POKE i,a
1080 NEXT i
1090 CALL &903C
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,F,90,C3,46,90,C3,87,90,C3,D4,90,46
1150 DATA 41,43,D4,41,4E,D0,43,4E,D0,0,0,0,0,0,0,0
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,4,90,21
1180 DATA 0,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22,1A,90,CD
1190 DATA 39,91,21,1E,90,CD,67,BD,7D,21,1E,90,11,23,90,F5
1200 DATA CD,54,91,F1,3D,FE,1,28,13,47,5,F5,C5,21,1E,90
1210 DATA 11,23,90,CD,79,BD,C1,F1,10,F1,18,DD,21,1E,90,ED
1220 DATA 5B,1A,90,CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91
1230 DATA DD,66,3,DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11
1240 DATA 37,90,CD,54,91,21,37,90,CD,4C,90,21,32,90,CD,8E
1250 DATA BD,21,2D,90,11,32,90,CD,79,BD,21,2D,90,CD,4C,90
1260 DATA 21,37,90,11,2D,90,CD,85,BD,21,37,90,ED,5B,1C,90
1270 DATA CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91,DD,66,3
1280 DATA DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11,37,90,CD
1290 DATA 54,91,21,37,90,CD,4C,90,21,32,90,11,28,90,CD,54
1300 DATA 91,21,28,90,CD,4C,90,21,32,90,CD,8E,BD,21,2D,90
1310 DATA 11,32,90,CD,79,BD,21,2D,90,CD,4C,90,21,28,90,11
1320 DATA 2D,90,CD,82,BD,21,37,90,11,28,90,CD,85,BD,21,37
1330 DATA 90,ED,5B,1C,90,CD,54,91,C9,11,1E,90,1,5,0,ED
1340 DATA B0,C9,11,2D,90,1,5,0,ED,B0,C9,11,32,90,1,5
1350 DATA 0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0

```

```
85 1C 0 B6 E2 95 F5 33 BF F5 4F FE A7 1 44 49 F2 4D E7 B7 91 D6
```

Version CPC 6128 :

```

1000 '-----
1010 ' Chargeur des RSX FACT, ANP et CNP
1020 ' Version CPC 6128
1030 '-----
1040 FOR i=&9000 TO &9159
1050   READ a$
1060   a=VAL("&" + a$)
1070   POKE i, a
1080 NEXT i
1090 CALL &903C
1100 '
1110 '-----
1120 ' Codes Op de la RSX FACT
1130 '-----
1140 DATA 0,0,0,0,F,90,C3,46,90,C3,87,90,C3,D4,90,46
1150 DATA 41,43,D4,41,4E,D0,43,4E,D0,0,0,0,0,0,0,0
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,4,90,21
1180 DATA 0,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22,1A,90,CD
1190 DATA 39,91,21,1E,90,CD,6A,BD,7D,21,1E,90,11,23,90,F5
1200 DATA CD,54,91,F1,3D,FE,1,28,13,47,5,F5,C5,21,1E,90
1210 DATA 11,23,90,CD,7C,BD,C1,F1,10,F1,18,DD,21,1E,90,ED
1220 DATA 5B,1A,90,CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91
1230 DATA DD,66,3,DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11
1240 DATA 37,90,CD,54,91,21,37,90,CD,4C,90,21,32,90,CD,91
1250 DATA BD,21,2D,90,11,32,90,CD,7C,BD,21,2D,90,CD,4C,90
1260 DATA 21,37,90,11,2D,90,CD,88,BD,21,37,90,ED,5B,1C,90
1270 DATA CD,54,91,C9,DD,66,1,DD,6E,0,CD,4B,91,DD,66,3
1280 DATA DD,6E,2,22,1C,90,CD,42,91,21,2D,90,11,37,90,CD
1290 DATA 54,91,21,37,90,CD,4C,90,21,32,90,11,28,90,CD,54
1300 DATA 91,21,28,90,CD,4C,90,21,32,90,CD,91,BD,21,2D,90
1310 DATA 11,32,90,CD,7C,BD,21,2D,90,CD,4C,90,21,28,90,11
1320 DATA 2D,90,CD,85,BD,21,37,90,11,28,90,CD,88,BD,21,37
1330 DATA 90,ED,5B,1C,90,CD,54,91,C9,11,1E,90,1,5,0,ED
1340 DATA B0,C9,11,2D,90,1,5,0,ED,B0,C9,11,32,90,1,5
1350 DATA 0,ED,B0,C9,1,5,0,ED,B0,C9,0,0,0,0,0,0

```

```
85 1C 0 B6 E2 98 F5 36 BF F5 52 2 AA 1 44 49 F5 50 ED B7 91 D6
```

Le programme en détail

Le programme débute par de nombreuses déclarations destinées à la manipulation de nombres réels :

FLOENT conversion flottant → entier ;
 ADDFLO addition de deux flottants ;
 NEGFOLO modification du signe d'un flottant ;
 MULFLO multiplication de deux flottants ;
 DIVFLO division de deux flottants.

Ces déclarations sont suivies des classiques données concernant les RSX. La table des RSX contient trois entrées :

```
Table : DB "FAC"
        DB "T"+80H → RSX ;FACT
        DB "AN"
        DB "P"+80H → RSX ;ANP
        DB "CN"
        DB "P"+80H → RSX ;CNP
        DB 0
```

La routine de définition des RSX est identique aux deux précédentes.

La RSX ;FACT est celle étudiée précédemment dans sa version réelle. Remarquez le point d'entrée interne EINT ligne 73 qui permet aux RSX ;ANP et ;CNP de faire directement appel à ;FACT.

Les RSX ;ANP et ;CNP s'expliquent d'elles-mêmes. Elles activent séquentiellement la routine de traitement de ;FACT pour calculer des quantités telles que $N!$, $P!$ ou $(N-P)!$. La routine de soustraction de flottants donnant parfois des résultats inattendus, nous avons préféré utiliser les routines NEGFOLO et ADDFLO pour réaliser les soustractions en flottant (lignes 141 et 144 dans la RSX ;ANP par exemple).

Application

Nombre de tiercés dans l'ordre pour 17 chevaux au départ :

```
n=17:p=3: ;ANP, @n, @p:PRINT n
```

Résultat :

4080

Nombre de tiercés dans le désordre pour 17 chevaux au départ :

```
n=17:p=3: ;CNP, @n, @p:PRINT n
```

Résultat :

680