

9/4.9.2

Calculatrice horaire

La plupart des calculatrices disponibles dans le commerce sont incapables d'additionner et de soustraire des données numériques horaires. Nous allons combler cette lacune en vous proposant un programme écrit en Basic et en Turbo Pascal. Ce programme est assez attrayant, puisque, contrairement à une calculatrice classique, il vous permet de visualiser les vingt-cinq dernières opérations.

COMMENT UTILISER LE PROGRAMME

Voici le listing de la version Basic du programme :

```

1000 ' -----
1010 ' Programme utilitaire pour additionner
1020 ' ou soustraire des heures/minutes/secondes
1030 ' -----
1040 '
1050 MODE 0
1060 LOCATE 8,1
1070 PRINT"H      M      S"
1080 LOCATE 5,2
1090 PRINT"-----"
1100 i=3
1110 '
1120 ' - - - - -
1130 ' Boucle de saisie
1140 ' - - - - -
1150 '
1160 '
1170 ' Saisie H
1180 '
1190 LOCATE 5,i
1200 INPUT "",h(i-2)
1210 LOCATE 5,i
1220 PRINT"      "
1230 LOCATE 5,i
1240 PRINT SPACE$(4-LEN(STR$(h(i-2)))) ;h(i-2)
1250 '
1260 ' Saisie M

```

```

1270 '
1280 LOCATE 10,i
1290 INPUT "",m(i-2)
1300 LOCATE 10,i
1310 PRINT"  "
1320 LOCATE 10,i
1330 PRINT SPACE$(4-LEN(STR$(m(i-2)))) ;m(i-2)
1340 '
1350 ' Saisie S
1360 '
1370 LOCATE 15,i
1380 INPUT "",s(i-2)
1390 LOCATE 15,i
1400 PRINT"  "
1410 LOCATE 15,i
1420 PRINT SPACE$(4-LEN(STR$(s(i-2)))) ;s(i-2)
1430 IF h(i-2)<0 THEN m(i-2)=-m(i-2):s(i-2)=-s(i-2)
1440 '
1450 ' Saisie operateur
1460 '
1470 LOCATE 2,i+1
1480 a$=INKEY$
1490 IF a$="" THEN 1480
1500 PRINT a$
1510 s$(i-2)=a$
1520 i=i+1
1530 IF a$<>"" THEN 1130
1540 '
1550 ' - - - - -
1560 ' Calcul du resultat
1570 ' - - - - -
1580 '
1590 LOCATE 2,i
1600 PRINT"  "
1610 LOCATE 5,i
1620 PRINT"-----"
1630 LOCATE 2,i+1
1640 PRINT"="
1650 hh=h(1)
1660 mm=m(1)
1670 ss=s(1)
1680 FOR j=2 TO i-3
1690 IF s$(j-1)="-" THEN 1740
1700 hh=hh+h(j)
1710 mm=mm+m(j)
1720 ss=ss+s(j)
1730 GOTO 1770
1740 hh=hh-h(j)
1750 mm=mm-m(j)
1760 ss=ss-s(j)

```

```

1770 NEXT j
1780 tot=hh*3600+mm*60+ss
1790 hh=FIX(tot/3600)
1800 mm=FIX((tot-hh*3600)/60)
1810 ss=tot-hh*3600-mm*60
1820 LOCATE 5,i+1
1830 '
1840 ' - - - - -
1850 ' Affichage du resultat
1860 ' - - - - -
1870 '
1880 PRINT SPACE$(4-LEN(STR$(hh)));hh
1890 LOCATE 10,i+1
1900 IF hh<0 THEN mm=ABS(mm)
1910 PRINT SPACE$(4-LEN(STR$(mm)));mm
1920 LOCATE 15,i+1
1930 IF hh<0 THEN ss=ABS(ss)
1940 PRINT SPACE$(4-LEN(STR$(ss)));ss
1950 END

```

Et la version Turbo Pascal du programme :

```

Program HMS;
{ ----- }
{ Programme utilitaire pour additionner      }
{ ou soustraire des heures/minutes/secondes }
{ ----- }

VAR
  I, J      : Integer;
  H, M, S   : Array[1..10] of Integer;
  S1        : String[10];
  a         : Char;
  si        : Array[1..10] of Char;
  HH, MM, SS : Integer;
  tot       : Integer;

begin
  ClrScr;
  GotoXY(8,1);
  Write('H   M   S');
  GotoXY(5,2);
  Write('-----');
  i:=3;

  { - - - - - }
  { Boucle de saisie }
  { - - - - - }

```

```

I:=3;
Repeat
  { ----- }
  { Saisie H }
  { ----- }
  GotoXY(5,i);
  Readln(h[i-2]);
  GotoXY(5,i);
  Write(' ');
  GotoXY(5,i);
  Str(H[I-2],s1);
  For J:=1 to 4-Length(s1) do
    Write(' ');
  Write(H[i-2]);

  { ----- }
  { Saisie M }
  { ----- }
  GotoXY(10,i);
  Readln(M[i-2]);
  GotoXY(10,i);
  Write(' ');
  GotoXY(10,i);

  Str(M[I-2],s1);
  For J:=1 to 4-Length(s1) do
    Write(' ');
  Write(M[i-2]);

  { ----- }
  { Saisie S }
  { ----- }
  GotoXY(15,i);
  Readln(s[i-2]);
  GotoXY(15,i);
  Write(' ');
  GotoXY(15,i);
  Str(S[I-2],s1);
  For J:=1 to 4-Length(s1) do
    Write(' ');
  Write(S[i-2]);
  IF (h[i-2]<0) THEN
  begin
    m[i-2]:=-m[i-2];
    s[i-2]:=-s[i-2];
  end;

```

```

{ ----- }
{ Saisie operateur }
{ ----- }
GotoXY(2,i+1);
Repeat
Until KeyPressed;
Read(Kbd,a);
Write(a);
si[i-2]:=a;
i:=i+1;
until (a=' ');

{ - - - - - }
{ Calcul du resultat }
{ - - - - - }
GotoXY(2,i);
Write(' ');
GotoXY(5,i);
Write('-----');
GotoXY(2,i+1);
Write('=');
hh:=h[1];
mm:=m[1];
ss:=s[1];
FOR j:=2 TO i-3 do
begin
  IF si[j-1]='+' THEN
    begin
      hh:=hh+h[j];
      mm:=mm+m[j];
      ss:=ss+s[j];
    end
  else
    begin
      hh:=hh-h[j];
      mm:=mm-m[j];
      ss:=ss-s[j];
    end;
end;
tot:=hh*3600+mm*60+ss;
hh:=TRUNC(tot/3600);
mm:=TRUNC((tot-hh*3600)/60);
ss:=tot-hh*3600-mm*60;
GotoXY(5,i+1);

```

```

{ - - - - - }
{ Affichage du resultat }
{ - - - - - }
Str(hh,s1);
For J:=1 to 4-Length(s1) do
  Write(' ');
Write(hh);
GotoXY(10,i+1);
IF (hh<0) THEN mm:=ABS(mm);
Str(mm,s1);
For J:=1 to 4-Length(s1) do
  Write(' ');
Write(mm);
GotoXY(15,i+1);
IF (hh<0) THEN ss:=ABS(ss);
Str(ss,s1);
For J:=1 to 4-Length(s1) do
  Write(' ');
Write(ss);
end.

```

Lancez le programme.

L'écran se présente comme suit :

```

      H      M      S
-----

```

Le curseur est positionné sous le champ **H**. Entrez une valeur pour ce champ et appuyez sur la touche **ENTER**. La donnée entrée est justifiée à droite du champ **H**, et le curseur se positionne sous le champ **M**. Entrez une valeur pour ce champ et appuyez sur la touche **ENTER**. La donnée entrée est justifiée à droite du champ **M**, et le curseur se positionne sous le champ **S**. Entrez une valeur pour ce champ et appuyez sur la touche **ENTER**.

Le curseur se positionne au début de la ligne suivante. Entrez le signe de l'opération à effectuer (+ ou -) et les trois champs du second opérateur. Recommencez ainsi jusqu'à ce que vous ayez entré toutes les opérations. Appuyez alors sur la touche **=**. Le résultat est immédiatement affiché.

Exemple :

```

      H      M      S
-----
      12      6      24
+      5      12      3
-      2      4      12
-----
=      15      14      15

```

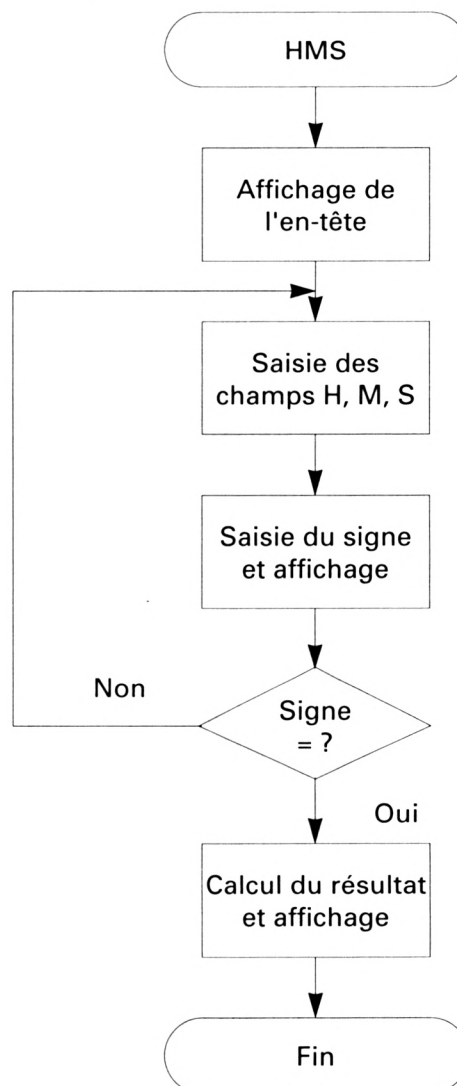
Attention :

La version Turbo Pascal ne fonctionne correctement que lorsque les calculs ne dépassent pas la valeur 32767 secondes. En effet, le programme utilise des variables entières. Toute valeur supérieure à cette limite est tronquée.

Ainsi, les opérations données en exemple ne donnent un résultat correct qu'en Basic. En effet, le résultat codé en secondes a pour valeur 54855, ce qui dépasse la limite supérieure.

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Nous allons analyser le fonctionnement de la version Turbo Pascal du programme. Vous n'aurez aucun mal à transférer les indications données à la version Basic.

Le programme est constitué de deux sections :

- la section de déclaration des variables ;
- la section du programme principal.

Dans la section de déclaration des variables, remarquez les tableaux **H**, **M**, **S** et **si** qui contiendront (respectivement) les champs heure, minute, seconde et le signe des opérations :

```
H, M, S : Array[1..10] of Integer;  
Si : Array[1..10] of Integer;
```

Dans le programme présenté dans le complément, le nombre de champs est limité à 10. Si vous possédez un CPC 6128, vous pourrez augmenter ce nombre sans aucun problème.

La section du programme principal débute par l'affichage de l'en-tête du programme :

```
ClrScr;  
GotoXY (8,1);  
Write('H      M      S');  
GotoXY (5,2);  
Write(' -----');
```

Les instructions suivantes représentent la boucle principale de saisie. Cette boucle est de type **Repeat Until**. Elle prend fin lorsque vous appuyez sur la touche "-" :

```
Repeat  
  ...  
Until (a='-');
```

La boucle de saisie peut être décomposée en quatre blocs :

- saisie du champ **H** ;
- saisie du champ **M** ;
- saisie du champ **S** ;
- saisie du signe de l'opération.

Les champs **H**, **M** et **S** sont saisis selon le même procédé.

Etudions la saisie du champ **H**.

Après le positionnement du curseur, une instruction **Readln** stocke la valeur numérique entrée dans le tableau **H** :

```
GotoXY(5,i);  
Readln(h[i-2]);
```


La valeur entrée est ensuite effacée :

```
GotoYX(5,i);
Write('  ');
```

puis réaffichée, justifiée à droite.

Pour connaître le nombre de chiffres entrés, le champ est converti en une chaîne, et le programme utilise la fonction **Length** :

```
Str(H[I-2], s1);
For J :=1 to 4-Length(s1) do
  Write(' ');
Write(H[I-2]);
```

L'opérateur est saisi d'une manière un peu particulière. Etant constitué d'un seul signe (+ ou -), il suffit d'utiliser la fonction **KeyPressed** pour l'acquérir :

```
Repeat
Until KeyPressed;
Read(Kbd,a);
```

Ce signe est affiché :

```
Write(a);
```

et mémorisé dans le tableau **si** :

```
si[i-2] :=a;
```

Lorsque la touche "=" est pressée, la boucle principale de saisie se termine, et le résultat est calculé.

Toutes les entrées du tableau **si** sont décrites :

```
FOR j :=2 TO i-3 do
begin
  IF si[j-1]='+' then
```

En fonction des valeurs testées, les données des tableaux **H**, **M** et **S** sont additionnées ou soustraites.

Lorsque toutes les données ont été passées en revue, le programme convertit les champs **H**, **M** et **S** du résultat en un nombre de secondes. Il redécompose ensuite ce nombre de secondes en des champs **H**, **M** et **S**. Cette méthode est destinée à éviter d'avoir des champs **M** ou **S** négatifs.

Les dernières lignes du programme affichent les champs **H**, **M** et **S** du résultat sur l'écran.

