

M.P.I. 5 LES COMMANDES D’AFFICHAGE.

1. AFFICHAGE DE TEXTE.

Il y a 2 façons d’afficher le texte qui semblent identiques, mais on emploie ces deux instructions dans de différentes situations.

1.1. INSTRUCTION DISP.

● Première approche:

```
: ClrHome  
: Disp 475  
: Disp "COUCOU"
```

Explications :

Efface l’écran
Affiche sur la ligne 1 à l’extrême droite le nombre 475
Affiche sur la ligne 2 à partir de la colonne 1 le texte entre guillemets

```
475  
COUCOU
```

Remarque.

Certains élèves peuvent avoir un affichage du nombre de type 475.00 et non pas entier. Il faut introduire dans le logiciel la notion d’affichage de la valeur numérique à savoir un chiffre entier et en mode normal pas scientifique ni ingénieur.

Les instructions employées se trouvent dans le menu MODE. On peut donc compléter notre programme par:

```
: ClrHome  
: Fix 0  
: Normal  
: Disp 475  
: Disp "COUCOU"
```

Explications :

Efface l’écran
La valeur numérique sera un nombre entier
En mode normal, ni scientifique ou ingénieur
Affiche sur la ligne 1 à l’extrême droite le nombre 475
Affiche sur la ligne 2 à partir de la colonne 1 le texte entre guillemets

● Autres possibilités:

```
: ClrHome  
: Disp 475, "COUCOU"
```

Ces 3 écritures différentes sont 3 méthodes différentes pour afficher la même chose. La méthode la plus classique est la seconde.

```
: ClrHome  
: Disp 475 : "COUCOU"
```

Avoir diverses possibilités d’écritures pour au final avoir le même résultat d’affichage, permet de développer différents styles de programmation. On peut ainsi avoir différents programmes, pour au final, avoir les mêmes applications.

● Pour afficher 475 à la 3^{ème} ligne et le coucou à la 4^{ème} ligne, on effectuera:

```
: ClrHome  
: Disp "", "", 475, "COUCOU"
```

Explications :

Efface l’écran
Affiche sur la ligne 3 à l’extrême droite le nombre 475 et sur la ligne 4 à partir de la colonne 1 le texte entre guillemets

```
475  
COUCOU
```

Disp "" établit juste une ligne blanche.

● Pour déplacer le texte sur une ligne.

On remarque que les nombres s’écrivent à droite de l’écran et le texte à gauche de l’écran. On peut déplacer le texte sur une même ligne en employant un artifice (5 espaces).

```
: ClrHome  
: Disp " COUCOU"
```

```
475  
COUCOU
```

1.2. INSTRUCTION OUTPUT.

Il n’existe pas de méthode pour déplacer vers la gauche les nombres. C’est une des raisons de l’utilisation de l’autre instruction

Output.

L’autre raison pour user de l’instruction **Output** à la place de **Disp** est, si l’on désire nombres et textes sur la même ligne.

```
: ClrHome  
: Output(4,3,475)  
: Output(5,6,"COUCOU")
```

Explications :

Efface l’écran
Affiche le nombre 475 à la ligne 4 et à la colonne 3.
Affiche le texte entre guillemets à la ligne 5 et à partir de la colonne 6.

```
475  
COUCOU
```

L’écran texte de la TI-83 est composé de 8 lignes et de 16 colonnes.

1.3. POUR CONCLURE.

Nous savons maintenant afficher un texte ou une valeur dans un programme. L’instruction **Disp** permet de faire apparaître texte et nombre à l’écran. L’instruction **Output** permet de faire apparaître texte et nombre à quelque endroit que l’on désire sur l’écran.

Par ailleurs, l’instruction **Disp** efface toute la ligne avant d’écrire le texte ou la valeur, alors que l’instruction **Output** écrit le texte ou la valeur à l’endroit indiqué et c’est tout.

Ainsi si on utilise pas l’instruction **Clrhome** avant l’instruction **Output**, et qu’il y a déjà des caractères inscrits sur cette même ligne de l’écran, on pourra se retrouver avec d’autres texte ou valeur avant ou après le texte ou la valeur qui vient d’être affiché.

2. UTILISATION DES VARIABLES.

Nous allons utiliser les variables que comporte la TI-83. Une variable est une lettre à laquelle on peut faire correspondre un chiffre.

La TI-83 compte 27 variables. Les lettres de A à Z plus teta

Cela est très utile pour comparer des valeurs ou pour rappeler un chiffre important à plusieurs endroit d'un programme.

```
: 1->A
: 2->B
: 10->H
: ClrHome
: Output(1,5,A)
: Output(3,5,B)
: Output(5,5,H)
: Stop
```

Explications :

Mémoire 1 dans la variable A
Mémoire 2 dans la variable B
Mémoire 10 dans la variable H

Affiche la valeur de A.

A n'est pas entre guillemets car on ne veut pas que la lettre A s'affiche à l'écran mais on veut sa valeur

Affiche la valeur de B

Affiche la valeur de H

Arrête le programme

Rien de bien difficile, cela reste utile est très important pour programmer en basic.

3. ENTRER DES DONNEES COMMANDE INPUT.

Nous allons voir comment rentrer des valeurs pendant l'exécution du programme, avec l'instruction **Input**. C'est une instruction très importantes car l'utilisateur du programme peut être impliqué: il participe activement, puisqu'on peut lui demander d'introduire des valeurs, du texte, des fonctions.... Il existe peu de **bons** programmes qui ne font pas appel à l'instruction **Input**, même dans le cas où on se limite à des questions qui font appel à des réponses du type Oui ou Non.

● Première possibilité:

```
: ClrHome
: Input A
: Disp A
: Stop
```

Explications :

Demande une valeur pour la variable A

Affiche la valeur de A, affichage comme pour un calcul.

● La plus courante pour demander une valeur à stocker (On peut utiliser une autre variable que A)

```
: ClrHome
: Input "Un entier A=",A
: Disp A
: Stop
```

Explications:

Affiche A

On affiche ce qui est entre guillemets à l'écran pour demander la valeur de la variable A.

(Dans l'exemple précédent un ? s'affichait à l'écran pour la saisie de A)

● Pour conclure:

Il n'est pas difficile de comprendre que cette dernière façon d'utiliser l'instruction **Input** est la plus courante, plutôt que la première approche car elle indique à l'utilisateur du programme ce qu'il doit entrer comme type de valeur. Car dans le premier cas, seul un ? est affiché comme unique demande impliquant l'entrée d'une valeur. Cela peut être une valeur, comme toute autre chose..... ce qui nécessite selon les cas, d'utiliser une certaine écriture pour entrer cette valeur dans le programme...

4. APPLICATION: QUELQUES PROGRAMMES

Nous allons appliquer ces quelques instructions dans la construction de petits programmes simples.

4.1. TRANSFORMER UNE DISTANCE KM EN U.A.

Consignes:

Je désire un programme qui va présenter successivement les écrans suivants:

Quelle est la
distance en km
?

Efface l'écran avant d'indiquer
la réponse

La distance
en unité
astronomique est
Le résultat avec deux chiffres
après la virgule

```

: ClrHome
: Normal
: Fix 2
: Output (4,3,"Quelle est la")
: Output (5,2,"distance en km")
: Disp "", "", "", "", ""
: Input C
: ClrHome
: Output (3,5,«La distance»)
: Output (4,4,«en unité»)
: Output (5,4,«astronomique est .....»)
: Output (6,7,C/150 000 000)
: Stop

```

Explications :

Affiche la valeur calculée en écriture «normale» (pas scientifique ni ingénieur)
 Fixe le résultat avec 2 chiffres après la virgule

4.2. REPARTITION ELECTRONIQUE.

Consignes:

Je désire un programme qui va présenter successivement les écrans suivants:

Valeur de Z
 comprise entre 2 et 10
 ?

*Efface l'écran avant d'indiquer
 la réponse*

La répartition
 électronique est:
 (K)2 (L) valeur

```

: ClrHome
: Normal
: Fix 2
: Output (4,3,"Valeur de Z")
: Output (5,2,"comprise entre 2 et 10")
: Disp "", "", "", "", ""
: Input Z
: Z - 2 -> L
: ClrHome
: Output (3,5,«La répartition»)
: Output (4,4,«électronique est»)
: Output (5,1,«(K)2»)
: Output (5,7,«(L)»)
: Output (5,11,L)
: Stop

```

Explications :

Affiche la valeur calculée en écriture «normale» (pas scientifique ni ingénieur)
 Fixe le résultat avec 2 chiffres après la virgule

Remarque:

Programme très simplifié, puisque pour le moment on ne peut pas donner la répartition électronique pour $Z < 2$ et $Z > 10$
 Il nous faut introduire la notion de condition... voir le chapitre suivant.

4.3. TRANSFORME UN NOMBRE N ENTIER EN ECRITURE HEURE, MINUTE, SECONDE.

On introduit une nouvelle commande, l'instruction iPart (dans le menu MATH - NUM), qui donne la partie entière d'un réel.

Exemple.

iPart (23,45) = 23

```

: ClrHome
: Normal
: Disp «Entrer le nombre de secondes»
: Input N
: iPart (N/3600) → H
: (N-H*3600) → R
: iPart (R/60) → M
: (R-M*60) → S
Disp «L'heure est»
Disp H, «Heure», M, «Minutes», S,
«Secondes»

```

Explications :

Affiche la valeur calculée en écriture «normale» (pas scientifique ni ingénieur)
 Demande à ce qu'on entre une valeur
 Qui sera enregistrée dans la variable N
 Donne la partie entière du calcul $N/3600$.