


## Principales commandes utiles à la réalisation d'un programme: TI

### Pour écrire, modifier ou exécuter un programme

Dans  choisir avec les touches directionnelles: soit le nom d'un programme, puis EXEC pour l'exécuter, soit NEW pour créer un nouveau programme, soit EDIT pour modifier le contenu d'un programme existant

### Pour saisir les entrées du programme

Après avoir fait NEW et donné un nom au programme, ou avoir choisi EDIT et le programme à modifier,

appuyer sur  puis avec les flèches choisir I/O ou E/S (entrées et sorties)


```
CTL I/O EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7↓getKey
```

on voit alors apparaître diverses commandes dont INPUT et PROMPT


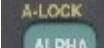
Il suffit alors , pour saisir une variable appelée A d'entrer « Input A » ou (et c'est surtout intéressant si on a plusieurs variables à affecter) « Prompt(A) »

Si on en a plusieurs on a le choix entre écrire « Input A » puis à la ligne « Input B » ou « Prompt (A,B) »

### Pour affecter une valeur à une variable:

Il faut utiliser la touche  de la calculatrice (to store: stocker, mettre en mémoire dans ce cas) qui affiche, à l'écran, une flèche)

### Pour que le programme affiche un nombre

Il suffit de demander « Disp A » si le nombre souhaité est A Disp se trouvant dans l'écran E/S ci-dessus  
Rq: Si on veut afficher du texte, par exemple *le nombre est* il faut écrire Disp « LE NOMBRE EST » en utilisant  puis  , ce qui bloque le clavier en fonction texte; les guillemets et l'espace se

trouvent aussi au clavier en mode texte.

### Pour écrire une condition avec « Si... alors... sinon...»

On trouve les commandes en appuyant sur  et en choisissant CTL:

```
CTL I/O EXEC
1:If
2:Then
3:Else
4:For(
5:While
6:Repeat
7↓End
```

On obtient alors ou en français: SI , ALORS, SINON, POUR( , TANT QUE,

REPETER, FIN...

On écrit en général:

« si *condition* alors *instruction* sinon *instruction* fin »

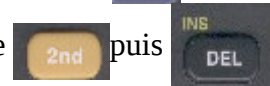
ex:

```
:■isp "COTE LE P  
LUS LONG"  
:Input A  
:Disp "AUTRES CO  
TES"  
:Prompt (B,C)  
:If A2=B2+C2  
:Then  
:Disp "RECTANGLE  
"  
:Else  
:Disp "PAS RECTA  
NGLE"  
:End
```

Le programme ci contre détermine si un triangle dont on connaît les longueurs est rectangle: On affecte trois nombres correspondant aux côtés du triangle dont on se demande s'il est rectangle, en commençant par celui qui peut être l'hypoténuse, du coup pour pouvoir écrire du texte avec *Disp* on utilise *Input*, puis pour les deux autres côtés, pour gagner du temps *Prompt* ).  
Si l'égalité de Pythagore est vérifiée, le triangle est rectangle (réciproque du Th) et le programme renvoie « RECTANGLE » sinon le programme renvoie « PAS RECTANGLE » (d'après la contraposée du Th)

Attention de bien respecter les sauts de ligne (avec ) et en cas

d'erreur pour insérer un caractère on utilise



### Pour écrire une boucle avec « tant que ... alors... »

While se trouve avec Si (ou If) et s'utilise ainsi:

« tant que *condition* passage à la ligne  
*instruction(s)* passage à la ligne  
Fin (ou End) »

ex:

```
:Prompt (A,B)  
:0→C  
:While A<B  
:A*1.1→A  
:B+2→B  
:C+1→C  
:End  
:Disp C
```

Dans le programme ci-contre, on demande deux nombres, *A* et *B* par exemple 500 et 600, puis on affecte à *C* la valeur 0.  
Ensuite tant que  $A < B$  on remplace *A* par  $1,1A$  et *B* par  $B+2$ , *C* augmentant de 1.  
A la fin, on affiche *C* (c'est à dire le nombre de fois qu'on a exécuté la boucle)  
Avec  $A=500$  et  $B=600$ , on a bien  $500 < 600$  donc *A* devient  $500 * 1,1 = 550$  et *B* devient 602, et *C* prend la valeur 1  
Comme  $550 < 602$  on refait la boucle, *A* prend alors la valeur  $550 * 1,1 = 605$  tandis que *B* devient 604 et *C* prend la valeur 2  
Comme  $605 > 604$  on ne refait pas la boucle, et on affiche la valeur de *C*: « 2 »

### Pour écrire une boucle avec « pour... de...à... »

On trouve *For* (ou *Pour*) avec *If* (ou *Si*) et *While* (ou *Tant que*)

On écrit: *For*( *nom de la variable*, *première valeur entière prise par la variable*, *dernière valeur entière prise par la variable*) c.a.d si on veut écrire *Pour K* allant de 0 à 9 on écrit: *For*( *K*,0,9)

ex:

```
:For(A,1,K)  
:(B+3)2→B  
:A+1→A  
:End
```

Ici l'extrait de programme va faire une boucle pour *A* allant de 1 à la valeur *K* qui doit être définie préalablement. Par exemple si  $K=3$  et si  $B=0$  au départ on aura:  
 $A=1$ , donc *B* devient  $(0+3)^2=9$  et *A* devient  $1+1=2$   
 $A=2$  donc on refait la boucle: *B* devient  $(9+3)^2=144$  et *A* devient  $2+1=3$   
 $A=3$  on fait une dernière fois la boucle: *B* devient  $(144+3)^2=21609$  et *A* devient  $3+1=4$  or *K* était égal à 3: on passe à la suite du programme.

### Pour trouver les symboles $>$ , $<$ , $=$ ... et les liens logiques: et, ou, ou exclusif, contraire

on utilise  puis



et ensuite avec les flèches on choisit « test » ou « logic »