

# Programmation avec une TI

Nous présentons ici un exemple d'algorithme traduit puis détaillons les différentes instructions existantes.

## *Un exemple de traduction d'algorithme*

Voici un algorithme simple qui calcule le périmètre et l'aire d'un rectangle en fonction de ses deux dimensions :

### *Variables*

les nombres longueur, largeur, aire, perimetre

### *Traitement*

*Demander* longueur

*Demander* largeur

aire  $\leftarrow$  longueur  $\times$  largeur

perimetre  $\leftarrow$  (longueur + largeur)  $\times$  2

*Afficher* aire

*Afficher* perimetre

Fin

et sa traduction en langage TI :

```
INPUT " LONGUEUR =", A
```

```
INPUT " LARGEUR =", B
```

```
A  $\times$  B  $\rightarrow$  C
```

```
(A+B)  $\times$  2  $\rightarrow$  D
```

```
DISP "AIRE =", C
```

```
DISP "PERIMETRE =", D
```

Remarques :

- la flèche d'affectation est dirigée vers la droite (ce qui est plus naturel : je mets la valeur  $\frac{1}{4}$  dans la variable  $\frac{1}{4}$ ) ;
- les variables ont des noms simples : A, B, C, etc. (ce qui complique la compréhension du programme) ;
- les textes à afficher sont entre guillemets ;
- il n'est pas nécessaire de faire la liste des variables au départ (de *déclarer* les variables).

## *Créer un nouveau programme*

Tapez sur la touche `PRGM` . Choisissez NEW.

Donnez un nom à votre programme (astuce : vous pouvez utiliser 2nd Alpha).

## *Entrée de valeurs dans un programme (Demander)*

Par exemple, pour demander une valeur à mettre dans la variable B :

```
PRGM I/O INPUT B ou INPUT " B " , B (alpha + pour ")
```

## *Mettre une valeur dans une variable (Affectation)*

Par exemple, pour mettre la valeur 3 dans la variable C : 3 `STO→` C

## *Afficher un résultat*

Par exemple, pour afficher le texte « Aire » suivi de la valeur contenue dans la

variable K : `PRGM I/O DISP 2nd alpha + A I`

```
R E + alpha , alpha K
```

# Programmation avec une TI

*Instruction conditionnelle (si condition alors instructions ¼)*

Exemple : Si A = 3 alors B = 4 sinon B = 5.

Faites bien la différence entre :

A = 3 qui est ici un test (A est-il égal à 3 ?)

B = 4 et B = 5 qui sont des instructions (affectations) à accomplir (ou pas suivant les cas).

Observez cette différence dans la programme en langage TI :

```
If A = 3
Then 4 → B
Else 5 → B
End
```

If, Then, Else, End :

PRGM

Control

Symboles =, < etc. :

2nd

MATH

## Boucles

Les boucles permettent de répéter des instructions un certain nombre de fois (ce nombre étant défini par une condition).

Les commandes liées aux boucles se trouvent dans

PRGM

Control

Il y a trois types de boucles :

Tant que condition vérifiée faire instruction(s)

Exemple : ajouter à un chiffre A (déjà connu) des 10 jusqu'à dépasser 100.

```
While A < 100
A + 10 → A
End
```

Remarque : l'instruction placée entre le While et le End n'est pas forcément exécutée si la condition n'est pas vérifiée.

Répéter instruction(s) jusqu'à ce que condition soit vérifiée

Même exemple que précédemment.

```
Repeat A < 100
A + 10 → A
End
```

Remarque : l'instruction est forcément exécutée au moins une fois car le test est fait à la fin de la boucle. La position de la condition (après le Repeat) rend cette boucle moins naturelle qu'une boucle avec un While.

Pour variable allant de val1 à val2 avec un pas de ¼ faire instructions  
Par exemple, calculer la somme des entiers pairs de 0 à 100.

```
0 → S
For (N,0,100,2)
S+N → S
End
Disp "SOMME = ", S
```