

# ALGORITHMIQUE - CASIO

Les touches indiquent les touches à taper et les AFFICHAGES indiquent les affichages écran.

## 1 Créer, éditer et exécuter un programme

### 1.1 Menu programmation

Touche **MENU**, choisir **PRGM** puis **EXEC** ou **EDIT** ou **NEW**.

- **NEW** permet de créer un nouveau programme :  
taper le nom du programme dans [ ], par exemple **HELLO**, puis **EXE**.
- **EDIT** permet d'écrire et corriger un programme :  
choisir le programme, par exemple **HELLO**, puis **EXE**.
- **EXEC** permet d'exécuter un programme :  
choisir le programme, par exemple **HELLO**, puis **EXE**.

### 1.2 Commandes de programmation

Dans l'éditeur de programme, vous avez accès aux commandes en appuyant sur :

- **SHIFT** + **VAR** → **PRGM** : pour les instructions de programmation (faire défiler avec **▶**)  
**COM** : If, For...      **REL** : <, >, =...      ?      ◀      :
- **OPTN** : pour les fonctions mathématiques (faire défiler avec **▶**)  
**PROB** : Ran#...      **Num** : Abs...
- **QUIT** ou **MENU** : pour quitter une application ou le programme

**Exemple 1.** Un premier programme qui affiche "HELLO WORLD"

```
= HELLO =  
"HELLO WORLD"
```

## 2 Les instructions simples

### 2.1 Les entrées

La commande **"A=": ?→A** affiche **A=** à l'écran, attend une réponse et stocke la valeur dans **A**.

La commande **?→A** n'affiche rien à l'écran, attend une réponse et stocke la valeur dans **A**

**Exemple 2.** Deux méthodes pour demander la valeur de A et B

```
"A=": ?→A ←
```

```
?→A ←
```

### 2.2 Les affectations

Pour affecter, c'est-à-dire stocker une valeur dans une variable, on utilise **→**

**Exemple 3.** Dans la variable A, stocker  $\frac{1}{4}$  et dans la variable B, stocker le résultat de  $A^2$  :

```
1/4→A  
A^2→B
```

### 2.3 Les sorties

La commande **"A=" ,A** affiche à l'écran **A=valeur\_de\_A**

**Exemple 4.** Affecter 5 à la variable A puis afficher : A=5 :

```
5→A ←  
"A=":A
```

```
// affiche A=
```

```
5
```

```
5→A ←  
"A=" ▲
```

```
// affiche A=
```

```
attend une validation par EXE
```

```
// affiche
```

```
5
```

## 3 La structure de programme

```
PROGRAM:EXEMPLE
```

```
"A=": ?→A ←
```

```
"B=": ?→B ←
```

```
100→N ←
```

```
A+B→S ←
```

```
bloc_d_instructions_si/pour/tantque
```

```
"W=":W ▲
```

## 4 Les blocs d'instructions : tests et boucles

### 4.1 Les instructions conditionnelles (ou tests) « Si » ou « If ».

```
Si ... alors ...
  If condition : Then action : IfEnd :
ou
  If condition ←
  Then action ←
  IfEnd ←
```

```
Si ... alors ... sinon ...
  If condition : Then action1 : Else action2 : IfEnd :
ou
  If condition ←
  Then action1 ←
  Else action2 ←
  IfEnd ←
```

**Exemple 5.** Par exemple pour stocker le maximum de A et B dans M :

```
If A>B ←
Then A→M ←
Else B→M ←
IfEnd ←
```

### 4.2 Les boucles « Pour » ou « For »

```
Pour I de A jusque B faire ...
  For A→I To B : action : Next :
ou
  For A→I To B ←
  action ←
  Next ←
```

```
Pour I de A jusque B pas P faire ...
  For A→I To B Step P : action : Next :
ou
  For A→I To B Step P ←
  action ←
  Next ←
```

**Exemple 6.** Calculer la somme des 100 premiers nombres entiers :

```
0→S ← // la somme S est initialisée à 0
For 1→I To 100 ←
S+I→S ← // dans la somme S, on stocke S augmenté de I (I variant de 1 à 100)
Next ←
S ▲
```

### 4.3 Les boucles « Tant que » ou « While »

```
Tant que ... faire ...
  While condition : action : WhileEnd :
ou
  While condition ←
  action ←
  WhileEnd ←
```

**Exemple 7.** Lancer un dé tant que l'on obtient pas 6 et compter le nombre de lancers nécessaire :

```
0→D ← // le dé D est initialisé à 0
0→C ← // le compteur C est initialisé à 0
While D<6 ←
Int(Ran#6)+1→D ← // dans le dé D, on stocke un nombre entier aléatoire entre 1 et 6
C+1→C ← // dans le compteur C, on stocke C augmenté de 1
WhileEnd ←
D ←
```

## Tableaux récapitulatifs des commandes

affectation	:a→2	dans la variable a, on <b>stocke</b> le nombre 2
entrée expression	:Input "A=",A	<b>afficher</b> « A= » à l'écran et <b>stocker</b> la réponse dans A
sortie	:Disp "A=",A	<b>afficher</b> « la valeur de a » à l'écran <b>afficher</b> « a=la valeur de a » à l'écran
arrêt dans une boucle		arrête la boucle
test/alternative	:If condition :Then :action :End	:If condition :Then :action1 :Else :action2 :End
boucle pour	:For(I,A,B) :action :End	:For(I,A,B,P) :action :End
boucle tantque	:While condition :action :End	

## Exemples de programmes Casio

**Exemple 8.** Calcul de la distance entre deux points  $A(x_A; y_A)$  et  $B(x_B; y_B)$ .

```
"xA=": ?→E ↵
"yA=": ?→F ↵
"xB=": ?→S ↵
"yB=": ?→T ↵
√((S-E)^2+(T-F)^2)→D ↵
"Distance=" ↵
D ◀
```

**Exemple 9.** Calcul du coefficient directeur et de l'ordonnée à l'origine de la droite passant par les points  $A(x_A; y_A)$  et  $B(x_B; y_B)$ .

```
"xA=": ?→E ↵
"yA=": ?→F ↵
"xB=": ?→S ↵
"yB=": ?→T ↵
(T-F)/(S-F)→A ↵
T-A*S→B ↵
"Y=AX+B" ↵
"A=": A ↵
"B=": B ◀
```

**Exemple 10.** Calcul des solutions de l'équation du second degré  $ax^2 + bx + c = 0$ .

```
"A=": ?→A ↵
"B=": ?→B ↵
"C=": ?→C ↵
B^2-4*A*C→D ↵
"DELTA=": D ◀
If D>0 ↵
Then "X1=": (-B-√(D))÷(2*A) ◀
"X2=": (-B+√(D))÷(2*A) ◀
Else If D=0 ↵
Then "X0=": -B(2*A) ◀
Else "PAS DE SOL" ◀
IfEnd ↵
IfEnd ↵
```

Remplir une liste (équivalent de la commande seq ?) : `listes` `OPT` `Remplir(`

