Algorithmique et programmation

Suites d'instruction

Exercice 1 Moyenne de trois nombres :

Algorithme

```
Lire A, B et C
A+B+C→S
S/3→M
Afficher M
```

Programme TI

```
Prompt A
Prompt B
Prompt C
A+B+C→S
S/3→M
Disp M
```

Programme Casio

```
"A="?→A ↵
"B="?→B ↵
"C="?→C ↵
A+B+C→S ↵
S/3→M ↵
M◢
```

Programme Python

```
A=input("A ?")
B=input("B ?")
C=input("C ?")
A+B+C→S
S/3→M
print M
```

Exercice 2 Ecrire un algorithme, et le programmer, qui demande les coordonnées de deux points A et B, et calcule et affiche les coordonnées du milieu I de $[AB]$ et la longueur AB.

Boucles itératives

Une boucle permet de répéter un ensemble d'instructions un nombre fixé de fois.

```
Pour variable de début à Fin
  instructions 1
  instructions 2
  ...
Fin Pour
```

Exercice 3 Affichage des carrés des 10 premiers entiers :

Algorithme

```
Pour I de 1 à 10
   Afficher I*I
Fin Pour
```

Programme TI

```
For (I,1,10)
Disp I*I
End
```

Programme Casio

```
For 1→I To 10 ↵
I*I◢
Next
```

Programme Python

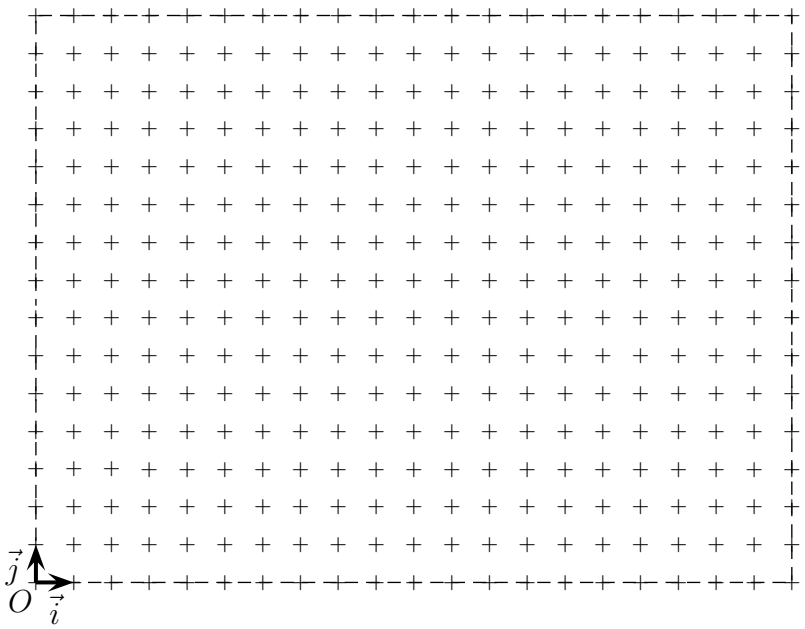
```
for i in range(1,11):
   print i*i
```

Exercice 4
Exécuter l'algorithme suivant sur le graphique ci-contre :

```
Pour i allant de 2 à 18
   Afficher le point (i;2)
   Afficher le point (i;10)
Fin Pour

Pour j allant de 3 à 9
   Afficher le point (2;j)
   Afficher le point (18;j)
Fin Pour

Pour i allant de 3 à 10
   Afficher le point (i;9+i/2)
   Afficher le point (i+8;15-i/2)
Fin Pour
```



Exercice 5 Calcul de la somme des carrés des 10 premiers entiers :

Algorithme

```
0→S
Pour I de 1 à 10
   S+I*I→S
Fin Pour
Afficher S
```

Programme TI

```
0→S
For (I,1,10)
S+I*I→S
End
Disp S
```

Programme Casio

```
0→S
For 1→I To 10 ↵
S+I*I→S ↵
Next
S◢
```

Programme Python

```
S=0
for i in range(1,11):
   print i*i
print S
```

Tests et instructions conditionnelles.

Un test est une comparaison entre la valeur d'une variable et une valeur donnée, ou entre les valeurs de deux variables.

Un test a deux résultats possibles : 0 (faux), ou 1 (vrai).
Les instructions ne sont effectuées que *si* le test indiqué est vrai.

```
Si test
   instructions 1
   instructions 2
   ...
Fin Si
```

Exercice 6 Test : nombre positif ?

Algorithme
```
Lire A
Si A>0
   Afficher "A positif"
Sinon
   Afficher "A négatif"
Fin Si
```

Programme TI
```
Prompt A
If A≥0
Then
Disp "A positif"
Else
Disp "A négatif"
End
```

Programme Casio
```
"A="?→A ↵
If A>0 ↵
Then "A positif" ↵
Else "A négatif" ↵
IfEnd
```

Programme Python
```
A=input("A ?")
if A≥0:
   print "A positif"
else:
   print "A negatif"
```

Exercice 7 Soit les points $A(2;3)$ et $B(-2;6)$. Ecrire un algorithme, et le programmer sur calculatrice, qui demande les coordonnées x et y d'un point M et affiche si M est plus près de A ou de B.

Exercice 8 Ecrire un algorithme, et le programmer, qui demande les coordonnées de trois points A, B et C, et qui indique, après calculs, si les points sont alignés ou non.

Boucles conditionnelles.

Une boucle conditionnelle permet de répéter une série d'instructions sans connaître a priori le nombre d'itérations.
La boucle est répété *tant que* le test indiqué est vrai.

```
Tant que test
   instructions 1
   instructions 2
   ...
Fin Tant que
```

Exercice 9 Compte à rebours :

Algorithme
```
Lire N
Tant que N>0
   N-1→N
   Afficher "N"
Fin Tant que
```

Programme TI
```
Prompt N
While N>0
N-1→N
Disp N
End
```

Programme Casio
```
"N="?→N
While N>1 ↵
N-1→N◢
WhileEnd
```

Programme Python
```
N=input("N ?")
While N>0:
   N=N-1
   print N
```

Exercice 10 Une ville compte 120 000 habitants. Sa population augmente de 4 % par an.
En combien d'années sa population aura-t'elle doublée ?
Ce nombre d'années dépend-il du nombre d'habitants initial ?

Exercice 11 4000 arbres poussent dans une forêt. Le nouveau plan d'exploitation prévoit, dès l'année prochaine, l'abattage de 20 % des arbres et la plantation de 1000 arbres chaque année.
Combien d'arbres cette forêt comptera-t'elle l'année prochaine ? Dans 3 ans ? Dans 10 ans ?
Le nombre d'arbres de cette forêt va-t'il se stabiliser, ou la forêt disparaître ?

Exercice 12 *Distributeur de billets*
Ecrire un algorithme qui demande un montant N en euros (un nombre entier) et qui calcule et affiche le nombre minimal de billets de 20 euros, de 10 euros et de 5 euros à fournir pour faire le montant N.

Jeu du nombre mystérieux. Ce jeu se joue à deux personnes de la manière suivante.

Un des deux joueurs choisit un nombre entier au hasard compris entre 1 et 100. Le but du deuxième joueur est de trouver ce nombre. Pour cela il propose un nombre au premier joueur qui lui fournit une des trois réponses :

- *Gagné*, si le nombre proposé est le bon ;
- *Trop grand*, si le nombre proposé est plus grand que le nombre mystérieux ;
- *Trop petit*, si le nombre proposé est plus petit que le nombre mystérieux ;

Si le nombre proposé n'est pas le bon, le deuxième joueur en propose un autre, et le jeu se poursuit, le but étant de trouver le nombre mystérieux le plus rapidement.

L'ordinateur fait deviner

On commence par le programme dans lequel l'ordinateur est le joueur choisissant un nombre au hasard compris entre 1 et 100, et l'utilisateur est le joueur qui doit trouver ce nombre.
Le programme doit aussi afficher le nombre de tentatives utilisées.

Lire attentivement l'algorithme suivant (ou un des programmes), bien suivre et comprendre la succession d'instruction, et indiquer le rôle de chacune des variables `M`, `C`, `N`.

Algorithme

```
M prend une valeur aléatoire entre 0 et 100
C prend la valeur 1
Afficher "Entrer un nombre"
Lire N
Tant que N≠M
   Si N<M alors afficher "Trop petit"
   Sinon afficher "Trop grand"
   Fin Si
   C prend la valeur C+1
   Afficher "Entrer un nombre"
   Lire N
Fin Tant que
Afficher "Gagne en",C," coups"
```

Programme TI

```
randInt(0,100)→M
1→C
Prompt N
while N≠M
if N<M
Then
Disp "Trop petit"
Else
Disp "Trop grand"
End
C+1→C
Prompt N
End
Disp "Gagne en",C," coups"
```

Programme Casio

```
Int(100×Ran#+1)→M↵
1→C ↵
"N="?→N ↵
While N≠M ↵
If N<M ↵
Then "Trop petit" ↵
Else "Trop grand" ↵
IfEnd ↵
C+1→C ↵
"N="?→N ↵
WhileEnd ↵
"Gagne" ↵
"Nombre de coups=" ↵
C◢
```

L'ordinateur doit deviner

Ecrire un programme pour ce jeu avec les rôles inversés : vous pensez à un nombre entier compris entre 1 et 100, et l'ordinateur doit le trouver.

Essayer de trouver une stratégie pour que l'ordinateur trouve ce nombre avec le moins de coups possible.