

Introduction à l'algorithmique et la programmation

Définition: Algorithme

Un algorithme est une succession d'instructions (aussi appelées commandes) et permettant la résolution systématique d'un problème donné.

Le mot "algorithme" vient du nom du mathématicien perse Al Khuwarizmi (latinisé au Moyen Age en Algoritmi), qui, au IXe siècle écrivit le premier ouvrage sur la résolution systématique de certaines équations (en plus d'introduire le zéro des Indiens).

Par exemple, l'algorithme suivant :

```
Pour A allant de 1 à 10
  stocker A*A dans B
  Afficher B
Fin Pour
```

permet d'afficher les carrés des 10 premiers entiers naturels.

Définition: Langage de programmation

Un langage de programmation est un ensemble d'instruction et de règles syntaxiques compréhensible par un système automatisé (calculatrice, ordinateur, puce électronique,...).

Un programme est alors la traduction d'un algorithme dans un langage de programmation particulier.

Il existe de très nombreux langage de programmation, par exemple, Basic, Fortran, C, C++, assembleur, Python, Matlab, ainsi que par exemple ceux implantés dans les calculatrices (alors dites "programmables"...).

Définition: Variable

On appelle variable tout emplacement de la mémoire dans lequel est, ou peut-être, stockée une information.

Une variable est constituée de :

- un nom qui permet à l'ordinateur de la localiser dans sa mémoire
- une valeur : l'information (souvent un nombre) qu'elle contient.

Tous les langages de programmation permettent d'effectuer des algorithmes comprenant des opérations élémentaires (addition, multiplication, division, ...) , ainsi que quelques structures fondamentales.

Suite d'opérations élémentaires :

```
Afficher "Entrer D: "
Lire D
R prend la valeur D/2
C prend la valeur 3,14*R*R
Afficher C
```

```
Lire A
Lire B
Lire C
D prend la valeur A+B+C
E prend la valeur D/3
Afficher E
```

Structure Tant que ..., faire ..., Fin Tant que (While ..., do ..., end)

```
Lire A
Tant que A>0 faire
  A prend la valeur A-1
  Afficher A
Fin Tant que
```

```
Lire N
Tant que N>0 faire
  N prend la valeur N-2
Fin Tant que
Afficher N
```

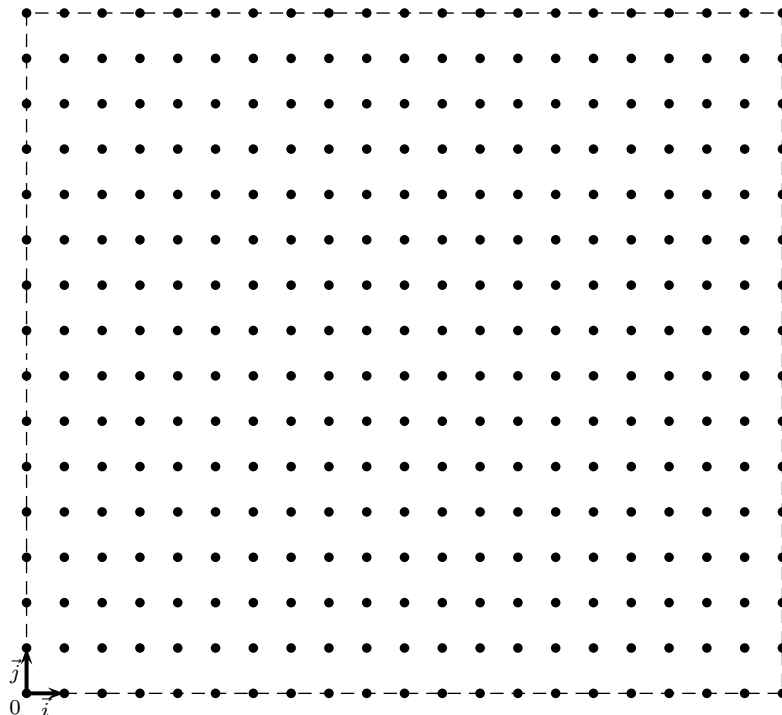
Structure (boucle) Pour *variable* allant de *début* à *fin*, faire ..., Fin Pour
(For ..., do ..., end)

Effectuer 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
```



Structure conditionnelle Si ... , faire ... , Sinon faire ... , Fin Si
(if ..., do ..., else do ..., end)

```
Afficher "Quel est votre age ?"
Lire A
Si A<18, alors
  Afficher "Vous êtes mineur"
Sinon
  Afficher "Vous êtes majeur"
Fin Si
```

```
Lire X1, X2, X3, X4, X5
M prend la valeur X1
Pour i allant de 2 à 5
  Si Xi > M, faire
    M prend la valeur Xi
  Fin Si
Fin Pour
Afficher M
```

Exercices : Ecrire un algorithme permettant de :

- demander à l'utilisateur les coordonnées de deux points A et B , et qui affiche les coordonnées du milieu I de $[AB]$ et la distance AB .
- demander à l'utilisateur les coordonnées d'un point M , et d'afficher, étant donné les points $A(3, 2; 0, 6)$ et $B(-1; 2)$, si M est sur la médiatrice de $[AB]$.
- Demander un nombre entier n et calculer la somme :
$$1 + 2 + 3 + \dots + n$$
- a) demander à l'utilisateur un nombre entier m et d'afficher la suite : 1 2 3 ... m
b) demander un nombre entier n et afficher les suites suivantes :
1
1 2
1 2 3
1 2 3 4
⋮
1 2 3 4 5 ... n