FICHE n°3 : DECOUVRIR



LES *BOUCLES*

Les boucles permettent de répéter des instructions.

- Boucle **Tant** **que** :

On peut répéter les mêmes instructions tant qu'une condition reste vérifiée.

En langage naturel, cela peut se présenter sous la forme suivante :

**Tant que** *Condition est vraie*

**Faire** *Instructions*

Exemple : **Tant que** *Le verre n'est pas plein*

**Faire** *Verser de l'eau*

- Boucle **Pour** :

On peut répéter les mêmes instructions pour un nombre de répétitions prédéfini par une variable.

En langage naturel, cela peut se présenter sous la forme suivante :

**Pour** *Variable* **allant de** *Valeur début* **à** *Valeur fin*

**Faire** *Instructions*

Exemple : **Pour** *Marche d'escalier* **allant de** *1* **à** *10*

**Faire** *Monter sur la marche suivante*

**Exercice 1 :**

On considère l'algorithme suivant donné en langage naturel :

**Entrée**

Saisir A

**Traitement et sortie**

Pour i allant de 1 à 5

Faire

A = A + 1

Afficher A

1. a) Pour A = 3 qu'affiche l'algorithme en sortie.

…………………………………………………………………………………………………………………..

b) Même question pour A = -4.

…………………………………………………………………………………………………………………..

1. Quelle valeur de A faut-il saisir pour obtenir en sortie l'affichage suivant :

-9 -8 -7 -6 -5

…………………………………………………………………………………………………………………..

**Exercice 2 :**

On considère l'algorithme suivant donné en langage naturel :

**Entrée**

Saisir *n*

**Traitement et sortie**

Tant que *n* < 50

Faire

*n* = *n* + 1

Afficher *n*

1. a) Pour *n* = 45 qu'affiche l'algorithme en sortie.

…………………………………………………………………………………………………………………..

b) Même question pour *n* = 48,1.

…………………………………………………………………………………………………………………..

c) Même question pour *n* = 53.

…………………………………………………………………………………………………………………..

1. Quelle valeur de *n* faut-il saisir pour obtenir en sortie l'affichage suivant :

44,3 45,3 46,3 47,3 48,3 49,3

…………………………………………………………………………………………………………………..

**Exercice 3 : La suite de Fibonacci**

On considère l'algorithme suivant donné en langage naturel :

**Initialisation**

Affecter à *n* la valeur 0

Affecter à A la valeur 1

Affecter à B la valeur 1

**Traitement et sortie**

Tant que *n* < 10

Faire

Affecter à *n* la valeur de *n* + 1

Affecter à C la valeur de B

Affecter à B la valeur de A + B

Affecter à A la valeur de C

Afficher B

1. Recopier et compléter le tableau par les valeurs successives prises par A, B et C.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *n* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| A | 1 |  |  |  |  |  |  |  |  |  |
| B | 1 |  |  |  |  |  |  |  |  |  |
| C | x |  |  |  |  |  |  |  |  |  |

1. Quel est l'affichage à la sortie de l'algorithme ?

…………………………………………………………………………………………………………………..

**Exercice 4 : Algorithme de tri**

On considère l'algorithme suivant donné en langage naturel :

**Entrée**

Saisir A, B, C

**Initialisation**

Affecter à *i* la valeur 0

**Traitement des données**

Tant que *i* < 2

Faire

Affecter à *i* la valeur de *i* + 1

Si A > B

Alors échanger les valeurs de A et B

Si B > C

Alors échanger les valeurs de B et C

**Sortie**

Afficher A, B, C

1. Tester l'algorithme pour différentes valeurs de A, B et C.

…………………………………………………………………………………………………………………..

1. Quel problème permet de résoudre cet algorithme ?

…………………………………………………………………………………………………………………..

**Exercice 5 :**

On considère l'algorithme suivant donné en langage naturel :

**Entrée**

Saisir N

**Initialisation**

Affecter à S la valeur 0

**Traitement des données**

Pour *i* allant de 1 à N

Faire

Affecter à S la valeur S + *i*

**Sortie**

Afficher S

1. Tester l'algorithme pour différentes valeurs de N.

…………………………………………………………………………………………………………………..

1. Quel problème permet de résoudre cet algorithme ?

…………………………………………………………………………………………………………………..

**Exercice 6 :**

Rédiger en langage naturel un algorithme permettant de calculer le produit d'une suite d'entiers naturels successifs, le premier terme et le dernier terme de cette suite étant saisis en entrée.

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………..