

TD N°3 : Les structures répétitives

Exercice 1

1. Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre **5**, le programme doit calculer:

$$1 + 2 + 3 + 4 + 5 = 15$$

2. Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

Exercice 2

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message: **Plus petit!**, et inversement, **Plus grand !** si le nombre est inférieur à 10

Exercice 3

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres:

Entrez le nombre numéro 1: 12

Entrez le nombre numéro 2: 14

etc.

Entrez le nombre numéro 20: 6

Le plus grand de ces nombres est: 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre:

C'était le nombre numéro 2

Exercice 4

Ecrire un algorithme permettant de lire une suite de nombres réels saisis au clavier et d'afficher la somme et le nombre des éléments lus. Le dernier élément à lire pour stopper la saisie est **Zéro**.

Exercice 5

Calculer a^b avec **a** réel et **b** entier par multiplication successives.

Exercice 6

Ecrire un algorithme qui lit un entier positif et vérifie si ce nombre est premier ou non.

Remarque : un nombre premier n'est divisible que par 1 et par lui-même (par exemple: **19**).

Exercice 7

Ecrire un algorithme qui lit deux entiers positifs A et B puis calcule et affiche leur PGCD en utilisant la méthode suivante:

- Si $A = B$; $\text{PGCD}(A, B) = A$
- Si $A > B$; $\text{PGCD}(A, B) = \text{PGCD}(A-B, B)$
- Si $A < B$; $\text{PGCD}(A, B) = \text{PGCD}(A, B-A)$

Exemple : $\text{PGCD}(18, 45) = \text{PGCD}(18, 27) = \text{PGCD}(18, 9) = \text{PGCD}(9, 9) = 9$

Exercice 8

Ecrire un algorithme qui calcule le PPCM (Plus Petit Commun Multiple) de deux entiers positifs A et B en utilisant la méthode suivante :

- Permuter, si nécessaire, les données de façon à ranger dans A le plus grand des deux entiers A et B;
- Chercher le plus petit multiple de A qui est aussi multiple de B.

Exemple :

$\text{PPCM}(6, 8) = \text{PPCM}(8, 6) =$ est ce que 8 est un multiple de 6 sinon.

On teste si 16, qui est $8+8$, est multiple de 6 si oui on s'arrête sinon on continue

Ensuite 24, qui est $16+8$, et puisque 24 est multiple de 6 on s'arrête 24.

Exercice 9

Ecrire un algorithme qui calcule et affiche les 100 premiers termes de la suite de Fibonacci.

La suite de Fibonacci est définie par:

- $F_0 = 1$
- $F_1 = 1$
- $F_n = F_{n-2} + F_{n-1}$ pour $n > 1$.

Exercice 10

Un nombre parfait est un nombre présentant la particularité d'être égal à la somme de tous ses diviseurs, excepté lui-même.

Le premier nombre parfait est $6 = 3 + 2 + 1$.

Ecrire un algorithme qui affiche tous les nombres parfaits inférieurs à 1000.