

TP N°5 : Reconnaître les bases (Traitement itératif)**Exercice 1 :**

Ecrivez un algorithme qui calcule la moyenne d'un étudiant : l'utilisateur saisira d'abord le **nombre** de notes, puis entrera les notes une à une.

Exercice 2 :

Ecrire un algorithme qui demande à l'utilisateur 10 entiers et qui indique le nombre des entiers positifs et le nombre des entiers négatifs.

Exercice 3 :

Ecrire un algorithme qui calcule et affiche la somme des entiers positifs compris entre 0 et un entier positif N entré par l'utilisateur.

Exercice 4 :

Ecrire un algorithme qui affiche les multiples de 3 se trouvant entre 3 et un nombre N supérieur ou égal à 3 entré par l'utilisateur.

Exercice 5 :

Ecrire un algorithme qui calcule et affiche la somme des entiers positifs impairs compris entre 0 et un entier positif N entré par l'utilisateur.

Exercice 6 :

Ecrire l'algorithme permettant de calculer la factorielle d'un nombre entier entré par l'utilisateur.

Exemple : $5! = 1*2*3*4*5=120$

Exercice 7 :

Ecrire un algorithme qui affiche la liste des diviseurs d'un entier positif N entré par l'utilisateur.

Exercice 8 :

Écrivez un algorithme qui demande à l'utilisateur un nombre entier positif, ensuite il détermine si ce nombre est parfait ou non.

Un entier N est dit parfait s'il est égal à la somme de ses diviseurs sauf le dernier qui est le nombre lui-même.

Exercice 9 :

Écrivez un algorithme qui demande à l'utilisateur un nombre entier positif, ensuite il détermine si ce nombre est premier ou non.

Un entier N est dit premier s'il est divisible par 1 et le nombre lui-même seulement.

Exercice 10 :

Ecrire un algorithme qui permet d'afficher la table de multiplication d'un nombre entier positif entré par l'utilisateur.

Exercice 11 :

Ecrire un algorithme qui permet d'afficher les tables de multiplication des nombres 2 à 9.

Chaque table se présentera comme suit :

Table des 5
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45

Exercice 12 :

Ecrivez un algorithme qui demande un entier positif n et un réel a puis calcule et affiche la somme suivante :

$$S_n(a) = 1 + a + a^2 + \dots + a^n = \sum_{k=0}^n a^k$$

Exercice 13 :

Ecrire l'algorithme qui calcule pour un entier positif N entré par l'utilisateur la somme suivante :

$$S(N) = 1 + 1/2 + 1/4 + 1/6 + \dots + 1/2*N, (N>0).$$

Exercice 14 :

Ecrire l'algorithme qui calcule pour un réel x et un entier N la somme suivante :

$$S(N) = 1 + x/(1!) + x^2/(2!) + \dots + x^N/(N!).$$

Exercice 15 :

Ecrire un algorithme qui calcule pour un entier positif N entré par l'utilisateur la somme :

$$S(N) = 1 + 1/3 + 1/9 + \dots + 1/3^N (N>0).$$

Exercice 16 :

Ecrire un algorithme qui calcule pour un entier positif N entré par l'utilisateur la somme :

$$S(N) = 1 - 1/2 + 1/3 - 1/4 + \dots \pm 1/N, (N>0).$$

Exercice 17 :

Ecrire un algorithme qui demande à l'utilisateur de lui fournir un nombre entier positif et inférieur à 100 et ceci jusqu'à ce que la réponse soit satisfaisante ; le dialogue se présentera ainsi.

Exemple d'exécution :

```
Donnez un entier positif inférieur à 100 : 452
Donnez un entier positif inférieur à 100 : 110
Donnez un entier positif inférieur à 100 : 28
Merci pour le nombre 28
```

Exercice 18 :

Ecrire un algorithme qui demande à l'utilisateur de saisir des entiers, puis affiche, après chaque saisie, le carré de la valeur saisie et qui interrompt la saisie si l'utilisateur entre la valeur -1.

Exercice 19 :

Ecrire un algorithme qui lit un nombre quelconque de valeurs entières en déterminant la somme des valeurs positives et la somme des valeurs négatives. On fait l'hypothèse qu'aucune de ces valeurs ne peut être nulle et que l'utilisateur introduira la valeur 0 pour signaler qu'il n'a plus de valeurs à fournir.

Exemple d'exécution :

```
Donnez un entier : 4
Donnez un entier : -8
Donnez un entier : -2
Donnez un entier : 3
Donnez un entier : 0
Somme des valeurs positives : 7
Somme des valeurs négatives : -10
```

Exercice 20 :

Ecrire un algorithme qui demande à l'utilisateur une série d'entiers, puis calcule leur somme en demandant à l'utilisateur à chaque fois s'il veut saisir un entier.

Exemple d'exécution :

```
Donnez un entier : 4
Voulez-vous ajouter un autre entier (O/N) : O
Donnez un entier : 1
Voulez-vous ajouter un autre entier (O/N) : O
Donnez un entier : 19
Voulez-vous ajouter un autre entier (O/N) : N
Somme des valeurs : 24
```

Exercice 21 :

Ecrivez un algorithme permettant d'afficher la table de multiplication d'un entier entré par l'utilisateur.

Modifier cet algorithme pour que l'on ne soit pas obligé de relancer le programme (proposer à l'utilisateur s'il veut continuer ou non).

Exercice 22 :

Ecrire un algorithme qui permet de Transformer un montant exprimé en DH en EURO (1 EURO = 10 DH à peu près).

Modifier le programme pour réaliser l'opération inverse aussi : (Conversion de l'EURO en DH).

Modifier le programme pour proposer à l'utilisateur un menu qui va lui permettre de choisir l'opération et boucler jusqu'à ce qu'un caractère particulier, par exemple 's' soit tapé.

Exercice 23 :

Programmer une petite calculatrice qui demande à l'utilisateur une opération à effectuer sous forme de caractère (par exemple '*', '+', etc.) Ou à l'aide d'un menu. Elle demande ensuite deux nombres réels, effectue le calcul demandé et affiche le résultat.

Le programme devra boucler jusqu'à ce qu'un caractère particulier, par exemple 's' soit tapé.

Exercice 24 :

Ecrire un algorithme qui demande à l'utilisateur un entier positif N, puis écrit une étoile, passe à la ligne, écrit deux étoiles, passe à la ligne, écrit trois étoiles... jusqu'à N étoiles afin d'obtenir ceci :

```
*
**
***
****
*****
.....
```

Exercice 25 :

Ecrire un algorithme qui demande à l'utilisateur un entier positif N, affiche un cône composé des étoiles qui s'étendant N lignes.

Exemple : N=5

```
      *
     ***
    *****
   ********
  *********
```