



Série d'exercice N° 03
Les structures Répétitives

Exercice 1 :

Écrire un programme qui affiche les diviseurs d'un entier positif N non nul

Exercice 2 :

Écrire un programme qui détermine si un nombre est premier ou non (Un nombre premier n'est divisible que par 1 et par lui-même)

Exercice 3 :

Écrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule Nième terme de la suite défini par :

$$U_n = \begin{cases} U_0 = 1 \\ U_n = 4U_{n-1} + 6 \end{cases}$$

Exercice 4 :

Écrire un programme qui demande à l'utilisateur de taper un entier N supérieur à 2, jusqu'à que la réponse convienne, puis qui calcul et affiche tous les termes de la suite de Fibonacci, inférieur ou égaux à N, la suite de Fibonacci est définie comme suite :

$$U_n = \begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} \end{cases}$$

Exercice 5 :

Écrire un programme qui affiche la table de multiplication pour les nombres 1 à 10

Exercice 6 :

Écrire un programme qui demande un nombre N, puis calcule et affiche la somme :

$$S = \sum_{i=1}^N 10^i$$

Exercice 7 :

Écrire un programme qui demande un nombre N, puis calcule et affiche la somme :

$$S = \sum_{i=1}^N \frac{1}{i}$$

Exercice 8 :

Écrire un programme qui demande un nombre positif non nul, et qui calcule sa factorielle

$$N! = \prod_{i=1}^N i$$

Exercice 9 :

Écrire un programme qui demande à l'utilisateur d'entrer un entier, puis le programme compte et affiche les chiffres qui composent cet entier

Exercice 10 :

Écrire un programme qui demande à l'utilisateur d'entrer un entier, puis le programme trouve et affiche l'inverse de ce nombre.

Exercice 11 :

Écrire le programme qui affiche la somme d'une suite d'entiers saisie par l'utilisateur se terminant par zéro.

Exercice 12 :

Écrire un programme qui demande un nombre N puis calcule :

$$S = \begin{cases} 1 + 3 + 5 + \dots + N & \text{si } N \text{ est impair} \\ 2 + 4 + 6 + \dots + N & \text{si } N \text{ est pair} \end{cases}$$

Exercice 13 :

Écrire un programme qui lit un nombre N , puis affiche si ce nombre est parfait ou non : un nombre est parfait s'il est égale à la somme de ses diviseurs

Exercice 14 :

Écrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne

Exercice 15 :

Écrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne

Exercice 16 :

Écrire un programme 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 17 :

Écrire un programme 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 18 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule le nombre de valeurs saisies. La saisie des nombres s'arrête lorsque l'utilisateur entre zéro

Exercice 19 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces nombres et quel était sa position. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 20 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro

Exercice 21 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 22 :

Écrire un algorithme qui : lit d'abord une valeur, ensuite il va lire successivement 10 nombres. Enfin il va déterminer combien de fois la première valeur a été saisie (sans compter la première saisie).

Exercice 23 :

Écrire un programme qui lit un réel X, puis calcule sa racine carrée en utilisant la suite suivante :

$$U_n = \begin{cases} U_0 = 1 \\ U_n = \frac{1}{2}(U_{n-1} + \frac{X}{U_{n-1}}) \end{cases}$$

On arrêtera les calculs lorsqu'on obtient une précision $|U_n - U_{n-1}| < 0.001$

Exercice 24 :

Écrire un programme qui lit le numérateur et dénominateur d'une fraction. Puis la simplifier

$$\frac{15}{45} = \frac{1}{3}$$

Exercice 25 :

Affichez une figure formée d'étoiles de N lignes (N est fourni au clavier):

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

Exercice 26 :

Affichez une figure formée d'étoiles de N lignes (N est fourni au clavier):

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

Exercice 27 :

Affichez une figure formée d'étoiles de N lignes (N est fourni au clavier):

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

Exercice 28 :

Affichez une figure formée d'étoiles de N lignes (N est fourni au clavier):

```

*****
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*****

```

Exercice 29 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*****
**           **
* *       * *
* * *   * *
*   **   *
*   **   *
* * *   *
* *       *
**           **
*****

```

Exercice 30 :

Écrire un programme permettant d'afficher les pyramides de nombres suivants :

```

1
121
12321
1234321
123454321
12345654321

```

Exercice 31 :

Le "jeu du nombre mystérieux" consiste à jouer contre l'ordinateur comme suit : L'ordinateur choisit, au hasard, un entier entre 1 et 100 et on doit le trouver en 7 essais au maximum grâce aux indices "C'est grand" et "C'est petit". Au moment venu on affichera "Bravo vous avez gagné !!". Si le nombre d'essais est atteint sans trouver le nombre mystérieux, le programme affichera alors "Perdu, le nombre cherché est : ", suivie du nombre à trouver. Écrire un programme pour s'amuser avec l'ordinateur.

Exercice 32 :

Écrire un programme qui détermine toutes les manières possibles d'obtenir un total de 15 en ajoutant trois entiers choisis entre 1 et 9.

Exercice 33 :

On se propose d'afficher un histogramme à l'aide des lettres A, B et C comme celui de l'exemple ci-dessous. Écrire un programme qui saisit le nombre de A, le nombre de B et le nombre de C puis affiche l'histogramme correspondant. Les nombres sont des entiers naturels inférieurs ou égaux à 15.

Exemple d'exécution : A =4,B = 7, C = 2

```

B
B
B

```

A B
 A B
 A B C
 A B C

Exercice 34 :

Écrire un programme qui permet de déterminer la somme des chiffres d'un nombre entier donné

exemple : pour $N = 25418$, on aura $2+5+4+1+8 = 20$

Exercice 35 :

Écrire un programme qui cherche et affiche tous les entiers cubiques de trois chiffres.

Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses trois chiffres.

Exemple : 153 est cubique car $153 = 1^3 + 5^3 + 3^3$

Exercice 36 :

On donne un entier naturel N strictement positif et on définit la suite de Syracuse par :

$$\text{suite de Syracuse : } \begin{cases} S_0 = N \\ S_{n+1} = \frac{S_n}{2} & S_n \text{ pair} \\ S_{n+1} = 3S_n + 1 & S_n \text{ impair} \end{cases}$$

Écrire un programme qui fait afficher les 50 premiers termes de cette suite.

Exercice 37 :

Si nous lançons 3 dés, le total des points est compris entre 3 et 18. Quelle est la probabilité d'avoir un total de 12 ?

Écrire un programme qui répond à cette question en simulant 100 lancers successifs.

Exercice 38 :

Pour un entier naturel N donné. Écrire un programme qui fait calculer et afficher la suite :

$$S = \sum_{i=1}^N \frac{1}{i!}$$

Exercice 39 :

Deux entiers m et n sont dit amis si et seulement si la somme des diviseurs de m sauf lui-même est égale à n et la somme des diviseurs de n sauf lui-même est égale à m . Écrire un programme permettant de déterminer et d'afficher tous les nombres amis compris entre 1 et 1000.

Exercice 40 :

Écrire un programme permettant de décomposer un entier N donné ($2 < N < 100$) en produit de facteurs premiers et d'afficher N et le produit de ses facteurs trouvés.

Exemple : Si $n = 60$ alors on affiche $60 = 2*2*3*5$.

Exercice 41 :

Sachant que :

$$\sin(x) = \sum_{i=1}^{\infty} \frac{x^i}{i!} \text{ tel que } x \in [0; 2\pi]$$

Écrire un programme qui permet d'afficher $\sin(x)$ en utilisant la formule ci-dessus. Le calcul s'arrête quand la différence entre deux termes consécutifs devient $< 10^{-4}$. La dernière somme calculée est une valeur approchée de $\sin(x)$.

Exercice 42 :

Un entier de n chiffres est dit bien ordonné si ses chiffres forment, de gauche à droite, une suite strictement croissante.

Écrire un programme qui saisit un entier N et fait sortir tous les entiers bien ordonnés de N chiffres et le nombre total de ces entiers.

Exercice 43 :

En mathématiques, un entier N est un carré parfait s'il existe un entier k tel que $N = k^2$.

Écrire un programme permettant de déterminer et d'afficher tous les carrés parfaits compris entre 1 et 9999

Exercice 44 :

Un entier de N chiffres ($1 < N < 9$) est dit bien ordonné si ses chiffres forment, de gauche à droite, une suite strictement croissante.

Exemple : L'entier de 3 chiffres, 147 est bien ordonné car $1 < 4 < 7$

Écrire un programme qui saisit un entier N ($1 < N < 9$) et fait sortir tous les entiers bien ordonnés de n chiffres et le nombre total de ces entiers.

Exercice 45 :

Écrire un programme qui affiche tous les entiers positifs impairs inférieurs à 100, en omettant les nombres divisibles par 7. L'affichage doit être sur 5 colonnes comme l'aspect suivant :

1	3	5	9	11
13	17	19	23	25
27	29	31	33	37
...

Exercice 46 :

En mathématiques, un entier N est un carré parfait s'il existe un entier k tel que $N = k^2$.

Écrire un programme permettant de déterminer et d'afficher tous les carrés parfaits compris entre 1 et 9999

Exercice 47 :

On se propose d'écrire un programme qui permet de déterminer et d'afficher toutes les représentations sous forme de sommes d'entiers consécutifs d'un entier n donné. Exemple d'exécution pour le nombre 45 :

$$45 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$$

$$45 = 5 + 6 + 7 + 8 + 9 + 10$$

$$45 = 7 + 8 + 9 + 10 + 11$$

$$45 = 14 + 15 + 16$$

$$45 = 22 + 23$$