

3 | Boucles et fonctions : Applications

1. Opérations de bases

1. Écrire une instruction permettant de vérifier si la partie imaginaire de $a = 2 + 5i$ est paire.
2. Écrire une instruction permettant de vérifier si la valeur contenue dans une variable b est un nombre entier.

2. Ecrire un programme qui vous demande un entier et affiche la table de multiplication de cet entier de 1 à 10.

Exemple :

```
>>> "n= ?" 4
1 x 4 = 4
2 x 4 = 8
3 x 4 = 12
4 x 4 = 16
5 x 4 = 20
6 x 4 = 24
7 x 4 = 28
8 x 4 = 32
9 x 4 = 36
10 x 4 = 40
```

3. Le refuge

Situation :

Vous souhaitez réserver une place dans un refuge pour préparer une randonnée mais souhaitez choisir le refuge dans lequel il y aura le moins de monde possible. Pour cela, vous téléphonez à tous les refuges, du plus proche (n°1) au plus éloigné (n°5) de votre point de départ, et leur demandez combien de personnes ils abritent déjà. Finalement, vous choisirez celui qui est à la fois le moins peuplé et le plus proche.

Entrées utilisateur :

Votre programme doit vous demander le nombre de personnes présentes dans les 5 refuges, numérotés de 1 à 5. La méthode d'affectation et de mise en mémoire du nombre de personnes est laissée libre.

Sortie :

Le programme indique le numéro du refuge qui contient le moins de monde et qui est le plus près ainsi que le nombre de personnes qui l'habitent.

Exemple :

```
>>> 'Nombre de personnes ?' 12 6 7 18 6
'Le refuge n°2 contient 6 personnes !'
```

Remarque :

L'utilisation des méthodes **.index()** et **min()** est interdite.

On rappelle qu'on accède au ième indice d'une liste L par l'instruction L[i], débutant à l'indice i=0

4. Les villages

Situation :

Lors d'une randonnée, vous souhaitez étudier le nombre de villages accessibles situés à moins de 30 km (inclus) de votre position actuelle par rapport au départ de la randonnée. Vous disposez d'une liste de 5 villages et de leurs positions par rapport au départ de la randonnée.

Entrée utilisateur :

Le programme vous demande votre position actuelle P. Par exemple P=20 indique que vous êtes à 20km du départ.

Le programme vous demande d'entrer les positions des 5 villages. Par exemple L=[12,128,50,73,81] indique les positions de 5 villages par rapport au départ. Le choix de la méthode d'affectation de ces positions est laissé libre (liste, affectation à 5 variables distinctes, n-uplet, ensemble, etc...)

Sortie :

Le programme indique combien de village sont situés à moins de 30 km (inclus) de votre position actuelle.

Exemple :

```
>>> 'P=?' 20
>>> 'L=?'
12
128
50
73
81
'Vous pouvez atteindre 2 villages'
```

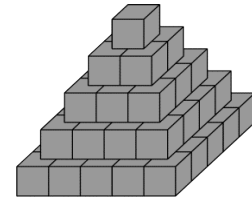
Amélioration :

Améliorer le programme de façon à ce qu'il puisse prendre en charge un nombre inconnu de villages à proximité (et donc sans se limiter à 5 villages).

5. La pyramide

Situation :

Vous décidez de réaliser une magnifique pyramide à partir de cubes que l'on vous propose selon le montage ci-contre. Mais avant de vous lancer dans sa construction, vous souhaitez savoir combien d'étages au maximum vous pourrez construire et combien de cubes vous seront nécessaires.



Entrée utilisateur :

Un nombre de bloc quelconque, mais positif.

Sortie :

La hauteur de la pyramide et le nombre de blocs utilisés pour la construire.

Exemple :

```
>>> 'Nombre de blocs ?' 37
```

```
Hauteur : 4          Blocs utilisés : 30
```

6. Le rectangle

Objectif :

Ecrire une fonction qui prend en paramètres un caractère (par exemple "X") et 2 entiers, largeur et hauteur, et qui représente un rectangle de dimension "largeur x hauteur" rempli du caractère choisi.

Exemple :

```
>>> rectangle("X",4,3)
XXXX
XXXX
XXXX
```

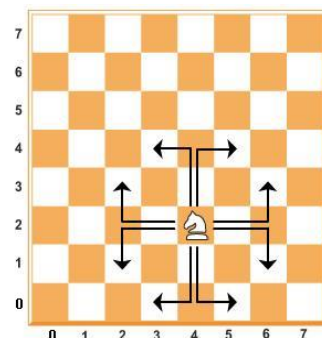
7. Le cavalier

Situation :

On dispose d'un cavalier sur un échiquier, placé à une position quelconque indiquée par l'utilisateur. On souhaite connaître les positions accessibles à ce cavalier.

Objectif et sortie :

Ecrire une fonction qui prend en paramètres les positions (x,y) initiales du cavalier et renvoie les positions accessibles par ce cavalier, sur l'échiquier. Les positions sur les axes sont numérotées de 0 à 7. Le cavalier ne doit pas sortir de l'échiquier. La forme choisie pour l'affichage de la sortie est laissée libre (affichage pas à pas, liste, n-uplet, autre ...).



8. Le cavalier et le pionSituation :

On place sur un échiquier un cavalier et un pion. On souhaite savoir si le cavalier peut atteindre le pion en 1 déplacement.

Objectif :

Ecrire une fonction prenant en paramètres la position initiale du cavalier et les coordonnées du pion. Cette fonction indiquera "Possible" si le pion peut-être atteint en 1 déplacement et "Impossible" sinon.