

Exercice 1

A partir d'une liste contenant des 'integers' (e.g., [1,2,4,5]), écrivez un programme Python qui retourne la somme des éléments de cette liste.

Exemple : [1,2,4,5] => 12

Exercice 2

A partir d'une liste contenant des 'integers', écrivez un programme Python qui retourne chaque élément de cette liste augmenté au carré.

Exemple : [1, 2, 3, 4] -> [1, 4, 9, 16]

Exercice 3

A partir d'une liste de 'string' (e.g., ['a', 'b', 'c']) et d'un séparateur (e.g., '|'), écrivez un programme Python qui permet de les joindre en un 'string'.

Exemple : pour la liste ['a', 'b', 'c'] et le séparateur '|' => résultat attendu : 'a|b|c'

Exercice 4

Cet exercice consiste à mettre en place un système qui permet à une petite boulangerie de gérer son stock.

Exercice 4a

Nous avons besoin d'un système qui demande répétitivement un 'sales keywords' au boulanger (afin que le boulanger puisse interagir avec les système). Si le boulanger entre 'exit', le programme doit se terminer. Sinon, le programme doit demander un autre 'sales keywords' (voir l'exemple)

Exemple :

```
Kindly enter a sales keywords: fdffd
Kindly enter a sales keywords: dsdasda
Kindly enter a sales keywords: exit
```

```
Process finished with exit code 0
```

Exercice 4b

Nous ajoutons maintenant une fonctionnalité qui permet au boulanger de gérer son stock. Il vend trois types de sandwich : saumon, salami et fromage. Au début de la journée, il vend 3 unités de chaque. Lorsque le boulanger entre un 'sales keywords', il y a trois cas de figure :

1. Il entre le nom d'un sandwich ('saumon', 'salami' ou 'fromage') qui est en stock => le programme réduit le stock d'une unité
2. Il entre le nom d'un sandwich mais il n'est plus en stock => le programme affiche « out-of-stock »
3. Le 'sales keywords' ne correspond pas au nom d'un sandwich => le programme affiche « invalid input »

Aide : utilisez un 'associate array' (également appelé 'dictionary') pour gérer le stock

Exemple :

```
Kindly enter a sales keywords: salami
Kindly enter a sales keywords: cheese
Kindly enter a sales keywords: abc
invalid input
Kindly enter a sales keywords: salami
Kindly enter a sales keywords: salami
Kindly enter a sales keywords: salami
out-of-order
Kindly enter a sales keywords: exit
invalid input
```

```
Process finished with exit code 0
```

Exercice 4c

Ajoutez une fonctionnalité qui permet, lorsque le boulanger entre `stock`, d'afficher l'état du stock pour chaque sandwich.

Exemple :

```
Kindly enter a sales keywords: stock
salami: 3
salmon: 3
cheese: 3
```

Exercice 4d

Ajoutez une fonctionnalité qui permet, lorsque le boulanger entre `analytics` d'afficher l'ordre dans lequel les sandwiches ont été vendus :

Exemple :

```
Kindly enter a sales keywords: salami
Kindly enter a sales keywords: test
invalid input
Kindly enter a sales keywords: cheese
Kindly enter a sales keywords: salami
Kindly enter a sales keywords: analytics
Today, sandwiches were sold in this order:
salami
cheese
salami
Kindly enter a sales keywords: exit
invalid input

Process finished with exit code 0
```

Exercice 5

Ecrivez un programme python qui permet d'imprimer tous les nombres impairs à partir de 1 jusqu'à un nombre n défini par l'utilisateur (qui doit être supérieur à 1) :

Exemple : si n = 6, résultat attendu : 1, 3, 5

Exercice 6 :

Ecrivez un programme python qui permet de dire si un nombre n entré par un utilisateur est un nombre premier. Est-il possible de réduire le nombre minimum d'itération pour parvenir à cela ?

Aide : l'opérateur modulo (%) retourne le reste de la division

Exemple : si n = 3, résultat attendu : « oui, 3 est un nombre premier »

Exercice 7 :

Considérez une liste énumérant les éléments entiers entre 1 et n, sachant que 2 éléments sont manquants. Par exemple, dans la liste [1, 2, 3, 5, 6, 8, 9, 10], les éléments 4 et 7 manquent. Ecrivez un programme python qui permet : 1) d'afficher les 2 éléments manquants et (2) d'insérer les deux éléments à l'index correct puis d'afficher cette liste.

Exemple : pour [1, 2, 3, 5, 6, 8, 9, 10], résultat attendu : [4, 7] [1,2,3,4,5,6,7,8,9,10]

Exercice 8 (difficile)

Considérer une liste de n 'integer' unique et positif, écrivez un programme python qui retourne toutes les permutations de cette liste.

Exemple : pour [1, 2, 3] n=3), résultat attendu : [1,2,3] , [1,3,2], [2,1,3], [2,3,1] [3,1,2] [3,2,1]

Exercice 9 :

Ecrivez un programme qui imprime la séquence de Fibonacci jusqu'au nombre n.

Exemple : si n=5 alors le programme doit imprimer : [0, 1, 1, 2, 3, 5]

Exercice 10 (difficile) :

Considérez un escalier avec n marches. Trouvez le nombre de possibilités unique d'aller de bas en haut de l'escalier, en considérant que vous pouvez faire 1 ou 2 marche(s) à la fois. Par exemple, si $n=3$, il y a trois façons possibles de monter l'escalier (voir l'image ci-dessous). Ecrivez un programme python qui calcule le nombre de façon possible de monter un escalier de n marches (considérant que $n>0$).

