

Q1 :

Programme Python qui affiche les nombres de 1 à 10.

Q2 :

Programme Python qui affiche tous les nombres pairs inférieurs à 100.

Q3 :

Programme Python qui affiche les nombres de 18 à 1 de trois en trois.

Exemple d'exécution :

```
18 17 16
15 14 13
12 11 10
9  8  7
6  5  4
3  2  1
```

Q3 :

Programme Python permettant d'afficher la table de multiplication d'un entier saisi par l'utilisateur.

Q4 :

Programme Python permettant d'afficher les tables de multiplications de tous les entiers de 1 à 9.

Affichage de trois tables de multiplication par ligne.

Q5 :

En utilisant la boucle for, Programme Python qui calcule et affiche la somme de N premier entiers positifs. N est un entier saisi par l'utilisateur.

Q6 :

Sans utiliser la fonction prédéfinie factorial(n) de la bibliothèque math qui calcule le factoriel d'un entier n, Ecrire un Programme Python qui calcule et affiche le factoriel d'un entier positif n saisi par l'utilisateur. Rappel : $n! = 1 * 2 * 3 * \dots * n$

Q7 :

Programme qui calcule la racine carrée de x (\sqrt{x}) est un nombre réel saisi par l'utilisateur.

Si le nombre saisi est négatif, le programme redemande de saisir un nouveau.

N.B: la fonction prédéfinie sqrt(x) de la bibliothèque math permet de calculer la racine carrée du nombre x.

Q8 :

Le plus grand diviseur commun (PGCD) de deux entiers positif a et b peut être calculer rapidement avec l'algorithme d'Euclide :

$\text{pgcd}(a, b) = \text{pgcd}(b, r)$ avec r est le reste de la division de a par b (ici on prend par exemple $a \geq b$)

On fait donc des divisions euclidiennes, jusqu'à ce qu'on trouve un reste nul. Le dernier reste non nul est le pgcd d'a et b.

Q9 :

Écrivez un programme qui trouvera tous ces nombres qui sont divisibles par 7 mais qui ne sont pas un multiple de 5, entre 2000 et 3200 (les deux inclus).

Les nombres obtenus doivent être imprimés dans une séquence séparée par des virgules sur une seule ligne.

Astuces :

Pensez à utiliser la méthode range(begin, end)

Q9 :

Écrivez un programme capable de calculer la factorielle d'un nombre donné.

Les résultats doivent être imprimés dans une séquence séparée par des virgules sur une seule ligne.

Supposons que l'entrée suivante soit fournie au programme :

8

Alors, la sortie devrait être :

40320

Indices : Si des données d'entrée sont fournies à la question, il faut supposer qu'il s'agit d'une entrée de console.

Q10 :

Avec un nombre entier donné n, écrivez un programme pour générer un dictionnaire qui contient (i, i*i) tel que est un nombre entier entre 1 et n (les deux inclus). et ensuite le programme doit imprimer le dictionnaire.

Supposons que l'entrée suivante soit fournie au programme :

8

Alors, la sortie devrait être :

{1 : 1, 2 : 4, 3 : 9, 4 : 16, 5 : 25, 6 : 36, 7 : 49, 8 : 64}

Astuces :

Dans le cas où des données d'entrée sont fournies à la question, il faut supposer qu'il s'agit d'une entrée de console. Pensez à utiliser dict()

Q11 :

Ecrivez un programme qui accepte une séquence de nombres séparés par des virgules depuis la console et génère une liste et un tuple contenant chaque nombre.

Supposons que l'entrée suivante soit fournie au programme :

34,67,55,33,12,98

Alors, la sortie devrait être :

['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

Astuces : Dans le cas où des données d'entrée sont fournies à la question, il faut supposer qu'il s'agit d'une entrée de console. La méthode tuple() permet de convertir une liste en tuple.

Q12 :

Définir une classe qui possède au moins deux méthodes :

getString : pour obtenir une chaîne de caractères de l'entrée de la console

printString : pour imprimer la chaîne en majuscules.

Veillez également inclure une fonction de test simple pour tester les méthodes de la classe.