Exercice premier pas avec python

1. Le présent de l'indicatif

Ecrire un programme qui demande à un utilisateur de saisir un verbe du premier groupe et affiche la conjugaison du verbe au présent de l'indicatif.

2. Distance entre deux mots

La distance de Hamming entre deux mots de même longueur est le nombre d'endroits où les lettres sont différentes.

Par exemple :

JAPON SAVON

La distance de Hamming entre JAPON et SAVON vaut donc 2.
Ecrire un programme qui demande à un utilisateur de saisir deux mots de même longueur et retourne la distance de Hamming. Idéalement si les deux mots saisie ne sont pas de même longueur elle retournera un message d'erreur.

3. Afficher un mot à l'envers

Demandez à l'utilisateur d'entrer un mot au clavier. Ensuite, écrivez-le à l'envers, c'est-à-dire de droite à gauche. Voici quelques conseils :

- Utilisez une boucle for avec indice, le nombre de lettres correspond aux nombres d'itérations.
- L'écriture mot[i] parcourt les caractères de gauche à droite. Ce n'est pas ce que nous voulons. Calculez à partir de la valeur de l'indice, et éventuellement de la longueur du mot, un autre indice partant de la droite et se déplaçant vers la gauche. Deux approches sont possibles, en utilisant une valeur d'indice positive ou négative.
- Affichez les caractères un à un en utilisant la fonction print(...,end='''') pour éviter les retours à la ligne.
- Pensez à faire des tests.

4. Calculer les solutions, si elles existe, du polynôme $ax^2 + bx + c = 0$

Demandez à l'utilisateur d'entrer un nombre quelconque. Attention, cela peut être un nombre à virgule. Calculez la valeur du polynôme et affichez-la. Redemandez deux autres valeurs et affichez les résultats.

- Utilisez la fonction input() pour lire au clavier.
- Pensez à convertir la chaîne de caractères retournée en nombre à virgule.
- Une boucle for peut vous éviter un copié-collé maladroit.

5. Additionner les n premiers entiers

L'utilisateur entre un nombre n au clavier. Vous devez calculer la somme des n premiers entiers : 1+ … + n, et l'afficher :

- Utilisez la fonction input() pour lire au clavier.
- Pensez à convertir la chaîne de caractères retournée en nombre entier.
- Utilisez une boucle for avec indice, son range dépend de la variable n.
- Créez une variable pour sommer les différentes valeurs. Pensez à l'initialiser correctement.
- Attention, un petit piège se cache sur la valeur de range, pensez à vérifier vos résultats.

6. L'infiniment petit

Vous allez diviser successivement par 2 la valeur 1, ceci 50 fois. À chaque fois, affichez le résultat sous la forme d'un nombre ayant 50 chiffres après la virgule. Pour valider votre résultat, affichez au démarrage la chaîne : "XX01234567890234567890123456789012345678901234567890123456789".

Voici un extrait de l'affichage que vous devez obtenir :

```
XX01234567890123456789012345678901234567890123456789
0.50000000000000000000000000000000000000000000000000
0.25000000000000000000000000000000000000000000000000
0.12500000000000000000000000000000000000000000000000
0.06250000000000000000000000000000000000000000000000
0.03125000000000000000000000000000000000000000000000
0.01562500000000000000000000000000000000000000000000
0.00781250000000000000000000000000000000000000000000
0.00390625000000000000000000000000000000000000000000
...
0.00000000000002842170943040400743484497070312500000
0.00000000000001421085471520200371742248535156250000
0.00000000000000710542735760100185871124267578125000
0.00000000000000355271367880050092935562133789062500
0.00000000000000177635683940025046467781066894531250
0.00000000000000088817841970012523233890533447265625
```

Quelques conseils :

- Utilisez l'expression "0123456789" * 5 pour dupliquer cinq fois cette chaîne et l'opérateur + pour concaténer "XX" au début.
- Pensez à utiliser la méthode .format() des chaînes de caractères.
- Utilisez l'opérateur de division standard /.

5. Compter les nombres à deux chiffres ayant le chiffre 7

Demandez à l'utilisateur d'entrer un nombre au clavier. Pour cet algorithme, voici quelques conseils :

- Le chiffre 7 peut être placé sur le chiffre des unités ou le chiffre des dizaines.
- Utilisez la fonction input() pour lire au clavier.

- Pensez à convertir la chaîne de caractères retournée en nombre entier.
- Créez une variable compteur et initialisez-la correctement.
- Grâce à une boucle for, parcourez l'ensemble des nombres à 2 chiffres, c'est-à-dire de 10 à 99. Pour cela, utilisez la fonction range(a,b), qui décrit une plage de valeurs partant de la valeur a et se terminant à la valeur b-1.
- Pour calculer le chiffre des dizaines, utilisez l'opérateur // (division entière).
- Pour calculer le chiffre des unités, retirez le chiffre des dizaines multiplié par 10.
- Affichez le résultat, vous devez trouver 18 !

En cas de souci sur un exercice, utilisez la fonction print() pour afficher ce qu'il se passe. Ici, vous pouvez facilement afficher les nombres détectés, cela permet de vérifier si tout est correct et de comprendre les problèmes.

6. Compter les consonnes dans un mot

Demandez à l'utilisateur d'entrer un mot au clavier. Pour cet algorithme, voici quelques conseils :

- Utilisez la fonction input() pour lire au clavier.
- Convertissez le mot en majuscules.
- Vous allez plutôt compter les voyelles !
- Créez une variable compteur et initialisez-la correctement.
- Parcourez chaque lettre du mot.
- Pour comparer deux lettres, utilisez l'opérateur == retournant True ou False.
- Écrivez une condition à base de plusieurs OU logique/ET logique pour détecter si la lettre correspond à une voyelle.
- Calculez et affichez le nombre de consonnes. Pensez à utiliser la fonction len() qui retourne le nombre de lettres dans le mot.

7. Encoder et décrypter un nom

Nous vous proposons d'encoder le nom d'un agent secret en utilisant le codage suivant : chaque lettre sera décalée de trois places dans l'ordre alphabétique. Ainsi, nous avons A→D, B→E, Y→B et Z→C. Demandez le nom de l'agent et affichez son nom encodé. Ensuite, demandez d'entrer un nom encodé et appliquez la méthode inverse pour le décoder. Voici quelques conseils :

- Demandez d'entrer un nom au clavier grâce à la fonction input().
- Convertissez automatiquement toutes les lettres en majuscules.
- Créez une chaîne vide pour stocker le résultat.
- Utilisez une boucle for pour parcourir chaque lettre du mot.
- Utilisez les fonctions de décodage et d'encodage Unicode vers un nombre entier : chr() et ord().
- Le caractère 'A' est codé 65 en ASCII. Pour obtenir le code de sa lettre correspondante 'D' égal à 68, vous pouvez faire 65 + 3. Il reste cependant à traiter le cas des dernières lettres X, Y et Z qui donnent A, B et C respectivement. Vous pouvez utiliser une condition pour traiter ces cas particuliers, plusieurs approches sont possibles.
- Utilisez l'opérateur + pour accoler le résultat courant à la nouvelle lettre.

- Affichez le résultat.
- Le décodage consiste à effectuer la même opération, mais dans l'autre sens ! Il n'y a pas de piège particulier.

8. Afficher les balises d'une page HTML

Le code HTML d'une page web est composé de balises ouvrantes <XXX> et fermantes </XXX>. Voici à quoi ressemble la structure minimale d'une page HTML :

<html> <head> <title> Mon Titre </title> </head> <body> Texte sur la page </body> </html>"

Créez un programme qui affiche toutes les balises présentes dans le code :

```
html
head
title
/title
/head
body
/body
/html
```

Voici quelques conseils :

- Le code de la page web étant stocké dans une chaîne unique, parcourez-la grâce à une boucle for.
- Recherchez le caractère "<" indiquant le début d'une balise.
- Lorsque le début d'une balise est détecté, recherchez la fermeture de la balise en recherchant le caractère ">" grâce à la fonction find(). Elle s'utilise depuis une chaîne de caractères en écrivant la syntaxe : phrase.find(). Pour éviter que la fonction find() démarre sa recherche au début, il faut lui indiquer en argument l'indice de départ de sa recherche : find(..., indice_depart).
- Une fois les positions de début et de fin détectées, affichez le contenu de la balise grâce à la syntaxe suivante : chaine[u:v]. Calculez judicieusement les valeurs de u et de v pour ne pas afficher les caractères < et >.

9. Afficher le calendrier du mois

Nous vous proposons d'écrire un calendrier respectant le format suivant :

```
LUN MAR MER JEU VEN SAM DIM
             1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30
```

Pour cela, en début de programme, l'utilisateur va entrer :

- Le nombre de jours dans le mois.
- Le numéro du jour démarrant le mois : 1 pour lundi,..., 7 pour dimanche.

Voici quelques conseils :

- Demandez à l'utilisateur d'entrer les valeurs au clavier grâce à deux appels à la fonction input(). Convertissez les chaînes de caractères récupérées en nombre entier.
- Il faut absolument créer une variable stockant la colonne courante. Si vous cherchez à chaque itération à recalculer la colonne courante en fonction de l'indice de boucle, cela va devenir TRÈS compliqué. Cette variable doit être initialisée avant la boucle for.
- Affichez la première ligne indiquant les jours.
- Le premier jour du mois moins 1 indique combien de colonnes sont vides en début de mois. Utilisez la syntaxe "␣␣␣␣"*n pour répéter n fois une chaîne de caractères donnée, car les colonnes vides font quatre caractères de large !
- Utilisez une boucle for unique sur le nombre de jours du mois.
- Pour éviter un retour à la ligne, utilisez l'argument end de la fonction print() : print(...,end='').
- Pensez à utiliser l'option format() dans la fonction print() pour que vos nombres se calent correctement dans les colonnes. Un nombre d'un ou deux chiffres doit être affiché sur trois colonnes et doit être suivi d'un espace.
- Après chaque affichage, la colonne courante devient la colonne à sa droite. Exception faite lorsque l'on quitte la colonne du dimanche, il faut repartir depuis la colonne du lundi. Pensez aussi dans cette configuration à faire un retour à la ligne.

Voici des exemples de l'exécution du programme :

```
>> Entrez le nombre de jours dans le mois :
>> 30
>> Entrez le premier jour du mois : 1 pour lundi, 7 pour dimanche
>> 3
>> LUN MAR MER JEU VEN SAM DIM
>>           1   2   3   4   5
>>   6   7   8   9  10  11  12
>>  13  14  15  16  17  18  19
>>  20  21  22  23  24  25  26
>>  27  28  29  30
>> Entrez le nombre de jours dans le mois :
>> 28
>> Entrez le premier jour du mois : 1 pour lundi, 7 pour dimanche
>> 7
>> LUN MAR MER JEU VEN SAM DIM
>>                           1
>>   2   3   4   5   6   7   8
>>   9  10  11  12  13  14  15
>>  16  17  18  19  20  21  22
>>  23  24  25  26  27  28
```

10. Statistiques avec pile ou face

Simulez un nombre donné de tirages pile ou face et calculez la proportion de chaque type de tirage.

Voici quelques conseils :

- Demandez à l'utilisateur d'entrer le nombre de tirages grâce à la fonction input(). Convertissez le résultat en nombre entier.

- Créez des variables permettant de compter le nombre de tirages pile et le nombre de tirages face. Pensez à les initialiser correctement.
- Simulez autant de tirages que demandé grâce à une boucle for avec indice. Utilisez la fonction : PileFace = random.randint(0,1). Cette fonction retourne aléatoirement la valeur 0 (pour pile) et 1 (pour face). Pour vous permettre d'accéder aux fonctions de tirages aléatoires du package random, la première ligne de votre programme doit être : import random.
- Calculez les pourcentage de tirages pile et face et affichez-les sous la forme d'un pourcentage présenté ainsi : 22 %. Vous pouvez utiliser la fonction format() pour afficher un nombre à virgule sur deux colonnes et sans chiffres après la virgule ou convertir le résultat en nombre entier pour l'affichage.