

Les exercices suivants sont le devoir n°1 du TD5 (Claudia NEGRI RIBALTA). Le nombre total de points est de 20. Vous devez travailler en groupes de 3.

Comme exigence de chaque exercice, le code doit être commenté (et bien commenté !) en expliquant ce que chaque morceau de code fait. Exemple : vous devez déclarer l'objectif du code, ce qu'elle fait, commenter chaque étape et dire si il fonctionne ou non. S'il n'y a pas de commentaire (significatif) le code ne sera pas noté et votre note sera pénalisée.

Il est également attendu que les variables soient nommées de manière compréhensible. Veuillez éviter les noms tels que n, x, y, n11, x12, e\_secondtry. Cela sera pénalisé.

Dans le cadre du devoir, pour les problèmes 3 et 4, un "flow chart" et une explication de la façon dont vous abordez le sujet peuvent être fournis dans le cadre des points supplémentaires. Je vous recommande vivement de le faire pour le problème 4, car il peut être compliqué. Ces deux éléments peuvent vous faire gagner un point supplémentaire si vous ne parvenez pas à obtenir une solution complète de l'exercice. Ces éléments doivent être inclus dans le dossier rendu, dans des fichiers différents.

**Veuillez envoyer le code et le diagramme de flux sous forme de zip, à Claudia.Negri-Ribalta@etu.univ-paris1.fr. Le nom du fichier doit être LASTNAME\_LASTNAME\_LASTNAME. Le titre de l'email doit être "Homework #1 Python : [Last\_names]". Ecrivez aussi les noms et prénoms de chaque personne du groupe dans le code (en commentaire) et dans l'email.**

**La date limite est le 01/03/2022 à 23:59. Si vous rencontrez des problèmes, veuillez me le faire savoir à l'avance.**

## Problèmes simples (4 points ; 0,5 bonus)

La série de problèmes suivante est très simple et ne devrait pas prendre beaucoup de temps.

1. (1 point) En utilisant la liste définie ci-dessous, trouvez le deuxième plus grand nombre et imprimez-le.

```
L = [13, 45, 34, 44, 104, 105, 1, 0]
# Votre code
# ...
print(second_maximum)
```

2. (0,5 points) Créez une fonction qui peut dire si l'année entrée par un utilisateur est une année bissextile ou non. Années bissextiles : 2400, 2000. Années non bissextiles : 2100, 1989.
3. (0,5 points) Créez un calculateur de valeur de pizza. Le programme doit demander à l'utilisateur la taille de la pizza (Petite, Moyenne, Grande), puis lui demander s'il veut du pepperoni et ensuite s'il veut du fromage. Enfin, le programme affiche à l'utilisateur le prix final.

La calculatrice doit imprimer à l'utilisateur le prix final en fonction de ses choix :

- (a) Petite Pizza : 15€
- (b) Pizza moyenne : 20€

- (c) Grande Pizza : 25€
- (d) Pepperoni pour petite pizza : +2€
- (e) Pepperoni pour une moyenne ou grande pizza : +3€
- (f) Supplément fromage pour toute taille de pizza : + 1€

Par exemple, une pizza Large (25€) avec Pepperoni (3€) mais sans fromage supplémentaire devrait coûter 28€.

4. (1 point) Demandez à l'utilisateur un nombre et, **à l'aide d'une fonction**, renvoyez le nombre (et imprimez-le !) avec des chiffres inversés. Le nombre peut être positif ou négatif. Par exemple, le nombre 435 en entrée donne 534 en sortie, et 534 en entrée donne -435 en sortie.
5. (1 point) Pour une phrase donnée écrite par l'utilisateur, renvoyez la longueur moyenne des mots. N'écrivez pas de caractères tels que des points ou autres. Si vous incluez la fonction permettant de supprimer les points, les virgules, les points d'exclamation ou les caractères spéciaux, vous gagnez 0,5 point supplémentaire.

## Générateur de mots de passe (3 points)

En utilisant le code fourni ci-dessous, créez un générateur de mots de passes aléatoires, de longueur choisie par l'utilisateur. En d'autres termes, l'utilisateur doit déclarer le nombre de caractères souhaité (par exemple : je veux qu'il fasse 10 caractères!) et le programme doit imprimer un mot de passe créé de manière aléatoire.

```
# Les bibliothèques utilisées

import string
import random

# caracteres pour generer le mot de passe
characters = list(string.ascii_letters + string.digits + "!@#%&*()")
```

Conseil ! Utilisez la bibliothèque random

## Jeu du pendu (4 points)

C'est l'heure du pendu ! Nous allons recréer le jeu avec des conditions spécifiques :

1. Il doit y avoir deux joueurs en compétition. Un(e) joueur(e) entre le mot et l'autre doit le deviner.
2. Il y a un nombre limité d'essais pour deviner un mot (définissons-le à 8).
3. Si le/la joueur(e) devine une lettre incorrecte, il/elle perd un essai.
4. Si un(e) joueur(e) veut deviner le mot, il/elle le peut. Mais s'il/elle se trompe, il/elle perd et le jeu est terminé. S'il/elle gagne, le jeu est également terminé.
5. Si toutes les lettres sont trouvées, le/la joueur(e) gagne.
6. Le jeu ne doit pas être sensible à la casse.
7. Le mot ne doit pas accepter d'espace mais peut accepter un caractère spécial. Si vous le souhaitez, vous pouvez activer une fonction pour désactiver les caractères spéciaux

```

\Welcome to Hangman game
Enter name of first player: Claudia
Enter name of second player: Sofia
Hello Sofia! Good luck guessing the word of Claudia
The game is about to start!
Let's play Hangman!
Claudia please select a random word: SanTiAgO
Now Sofia you need to guess the word. You have 8 tries for this.
You can either guess a letter or a word. But remember! If you guess the wrong word, you loose!
['_', '_', '_', '_', '_', '_', '_', '_']

Amount of attempts left: 8

Please write the letter or word to guess: A
Yes! That letter is inside the word!
['_', 'a', '_', '_', '_', 'a', '_', '_']

Amount of attempts left: 8

Please write the letter or word to guess: |

```

---

Figure 1: Exemple de jeu de pendu

## Jeu du morpion (9 points)

Nous allons travailler avec le jeu du morpion cette fois-ci ! Les règles du jeu sont les suivantes

1. Il doit y avoir deux joueurs (0,5 point)
2. Le plateau doit être imprimé dans la console (2,5 points)
3. Un(e) joueur(e) doit être le "O" et l'autre le "X" (1 point)
4. L'algorithme doit garder la trace de l'endroit où chaque joueur a placé son X/O (2 points). Si un espace est déjà occupé, il ne peut pas être réécrit (1 point)
5. Le jeu doit vérifier les conditions de victoire (trois marques identiques alignées verticalement, horizontalement ou en diagonale), et qui a gagné. (2 points)

```

Player X turn
- - -
- - -
- - -
Enter row and column numbers to fix spot, with a space between the numbers: 2 2

Player O turn
- - -
- X -
- - -
Enter row and column numbers to fix spot, with a space between the numbers:

```

---

Figure 2: Exemple de jeu de morpion