**TD 2**

**LA BOUCLE BORNEE, LES LISTES**

**ET PARCOURS DE LISTE**

**PARTIE I - LE CHOIX**

**Exercice 1.1 :**

Partie A : Ecrire un programme en Python qui calcule le volume d'un cylindre.

Entrée : pi, rayon, hauteur du cylindre

Sortie : volume du cylindre

Calcul : volume = pi x rayon2 x hauteur

Volume cylindre = π x r2 x H

Partie B : Ajouter la possibilité de calculer le volume d’un cube au choix de

l’utilisateur.

*Exemple d’interface utilisateur* :

Voulez vous calculer le volume d’un cube (ca) ou d’un cyclindre (cy) ? ca

Entrez le côté du cube : 10

Le volume du cube est de 100

**Réponse :**

choix = input("Voulez vous calculer le volume d’un cube (ca) ou d’un cyclindre (cy) ?")

# Partie A

if choix == "cy":

pi = 3.14

hauteur = int(input("Donnez la hauteur du cylindre."))

rayon = int(input("Donnez le rayon du cylindre."))

volume = pi \* rayon\*\*2 \*hauteur

# Partie B

if choix == "ca":

coté = int(input("Donnez le coté du cube."))

volume = coté\*\*3

print(volume)

**Exercice 1.2 :**

Vous allez écrire un programme en Python qui calcule le prix d’une place de cinéma en fonction de l’âge.

* si l’utilisateur a en dessous de 12 ans il paie 4 euros la place
* s’il a entre 12 et 18 ans, il paie 5,5 euros
* s’il a au-dessus de 18 ans il paie plein tarif soit 7,5 euros sauf s’il est étudiant et qu’il a moins de 26 ans, dans ce cas il paie 5,50 euros
* s’il a au-dessus de 65 ans il paie 5,5 euros

1. Ecrivez en langage naturel l’algorithme de cet exercice

Age <- Affecter un entier  
Prix <- Initier à 0

Etudiant <- Demander o ou n  
Si age < 12 :

Affecter prix à 4

Si age >= 12 et age <= 18:

Affecter prix à 5.5

Si age > 65 :

Affecter prix à 5.5  
Si age > 18 :

Affecter prix à 7.5

Si etudiant = o et age < 26 :

Affecter prix à 5.5

Afficher prix

1. Ecrivez l’organigramme correspondant sur feuille et appelez votre professeur pour vérification
2. Implémentez en Python l’algorithme

Réponse :

age = int(input("Bonjour, veuillez donner votre âge s'il vous plait!"))

prix = 0

etudiant = input("Etudiant? (o ou n)")

if age <=0 :

print("Oula… Vous y allez de main morte")

else :

if age < 12:

prix = 4

elif age >= 12 and age <= 18:

prix = 5.6

elif age > 65:

prix = 5.5

elif age > 18:

prix = 7.5

if etudiant == "o" and age < 26:

prix = 5.5

print(prix)

**PARTIE II - LA BOUCLE BORNEE ET RANGE()**

**Exercice 2.1 :**

Vous allez écrire un programme qui affiche les nombres divisibles par 5 parmi les 100 premiers nombres .

1. Que signifie divisible par 5 ? La division euclidienne par 5 a un reste nul.
2. Quel opérateur arithmétique permet de répondre à la question 1 ? % (modulo)
3. Implémentez en Python le programme qui permet de répondre à la problématique posée

**Réponse :**

for i in range(101):

divisible = i%5

if divisible == 0:

print(i, end = " ")

**Exercice 2.2 :**

Vous allez calculer la factorielle d’un nombre n donnez par l’utilisateur.

1. Cherchez et écrivez la définition d’une factorielle

La factorielle d'un entier naturel n est **le produit des nombres entiers strictement positifs inférieurs ou égaux à n**.

1. Ecrivez en langage naturel l’algorithme de calcul d’une factorielle

Factorielle <- 1  
Demander n (entier)  
Pour i allant de 1 à n+1 :  
 Affecter i\*factorielle à factorielle  
Afficher factorielle

1. Implémentez en Python l’algorithme décrit ci-dessus

**Réponse :**

factorielle = 1

n = int(input("Donnez un nombre."))

for i in range(1, n+1) :

factorielle \*= i

print(factorielle)

**PARTIE III - LES LISTES ET BOUCLES DE PARCOURS DE LISTE**

**Exercice 3.1** :

A partir d’une liste de données alphanumériques, on souhaite rechercher la présence d’un élément donné par l’utilisateur.

1. Expliquez brièvement quel type de boucle on va utiliser et pourquoi   
   On utilise **for** pour parcourir l’ensemble des éléments de la liste.
2. Ecrivez en langage naturel l’algorithme que vous proposez  
   liste <- liste non vide  
   Demander recherche  
   verif <- Faux  
   Si recherche dans liste :  
    Affecter Vrai à verif  
   Afficher verif
3. Implémente en Python l’algorithme précédent

**Réponse :**

liste = ['a', 'b', "3", "69"]

recherche = input("Recherchez un élément")

verif = False

if recherche in liste :

verif = True

print(verif)

**Exercice 3.2 :**

Recherche d’un minimum et d’un maximum dans une liste.

Partie I : écrivez un programme en Python qui demande à l’utilisateur d’entrer les moyennes d’une classe de 10 élèves.

Partie II : écrivez la suite du programme qui renvoie la meilleure moyenne et la moins bonne moyenne contenue dans la liste. Pour cela :

1. Expliquez en langage naturel comment trouver la valeur minimum dans une liste  
   liste <- liste vide

Pour i allant de 0 à 10 :

Demander moyenne  
 Ajouter moyenne à liste  
Compare = 0  
minimum = 0  
maximum = 0  
Pour tous éléments dans liste :  
 Si element supérieur à compare :  
 compare <- element  
 maximum <- element  
Pour tous éléments dans liste :  
 Si element inferieur à compare :

compare <- element  
minimum <- element

Afficher minimum et maximum

1. Implémentez votre algorithme dans un programme Python

**Réponse :**

liste = []

for i in range(11):

moyenne = float(input("Moyenne de l'élève"))

if moyenne <0 or moyenne >20:

print("Vous notez bizarrement...")

break

liste.append(moyenne)

compare = 0

min = 0

max = 0

for element in liste:

if element > compare:

compare = element

max = element

for elements in liste:

if elements < compare:

compare = elements

min = elements

print(min, max)

**PARTIE IV - LES LISTES, BOUCLE BORNEE ET BOUCLE DE PARCOURS DE LISTE**

**Exercice 4.1 (suite du 3.2) :**

Partie III : ajoutez dans la partie I la possibilité d’entrer le nom de l’élève dans une deuxième liste, en même temps que la moyenne.

*Exemple d’interface utilisateur* :

Nom de l’élève ? patrick

Entrez sa moyenne : 11.2

Partie IV : vous afficherez la meilleure moyenne, ainsi que la pire moyenne, avec le nom de l’élève qui leur correspond.

**Réponse :**

listem = []

liste\_nom = []

for i in range(4):

moyenne = float(input("Moyenne de l'élève"))

nom = input("Nom de l'élève")

listem.append(moyenne)

liste\_nom.append(nom)

if moyenne <0 or moyenne >20:

print("Vous notez bizarrement...")

break

compare = 0

min = 0

max = 0

nom\_min = ""

nom\_max = ""

for element in range(len(listem)):

if listem[element] > compare:

compare = listem[element]

max = listem[element]

nom\_max = liste\_nom[element]

for elements in range(len(listem)):

if listem[elements] < compare:

compare = listem[elements]

min = listem[elements]

nom\_min = liste\_nom[elements]

print(nom\_min, ":", min, ",", nom\_max, ":", max)

**PARTIE V - IMPLEMENTER UN ALGORITHME QUE JE N’AI PAS ECRIT**

**Exercice 5.1 :**

Voici l’algorithme en langage naturel d’un tri à bulle. Implémentez le en Python avec la liste suivante : liste = [19, 22, 1, 14, 8, -3, 31, 100, -7, 55]

pour i variant de 1 à longueur\_liste

pour j variant de 1 à longueur\_liste - i – 1

si liste [j] > liste [j+1]

échanger(liste [j], liste [j+1])

tip 1. La longueur d’une liste s’obtient grâce à la fonction len(). Par exemple len(liste) renverra ici 10, c’est-à-dire le nombre d’éléments qu’elle contient.

tip 2. L’échange de deux valeurs peut s’écrire en Python : a, b = b, a

**Réponse :**

liste = [19, 22, 1, 14, 8, -3, 31, 100, -7, 55]

for i in range(1, len(liste)):

for j in range(len(liste)-i):

if liste[j] > liste[j+1]:

liste[j], liste[j+1] = liste[j+1], liste[j]

print(liste)