

Module : Atelier de programmation

Enseignant : Dr Ahmed Rebai

Classes : MBA - 1

Documents autorisés : Oui, Examen théorique et pratique sur Anaconda (PC)

Il est strictement interdit de se connecter sur internet pendant l'examen.

Nombre de pages : 4 pages

Date : Mercredi le 07/11/2018 Heure: 09h:00mn Durée : 2H00.

Il est attendu que les réponses fournies soient clairement justifiées.

La clarté, la rédaction et la justification des réponses fournies interviennent dans la notation.

Si, au cours de l'examen, un étudiant repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa rédaction en expliquant les raisons des initiatives qu'il est amené à prendre.

L'étudiant doit spécifier la technologie utilisée pendant l'implémentation (R, C, C++, Java ou Python). Dans le cas échéant, il peut aussi ajouter l'algorithme en français.

Rappels :

- Pour éviter toute erreur de comptage (un n transformé en $n + 1$ ou $n - 1$), il est indispensable de vérifier le bon fonctionnement du programme sur de petites valeurs de n.

- On sera attentif aux indentations (décalages).

- Une fonction Python qui accepte x en entrée et renvoie y en sortie s'écrit

def f(x) :

 corps de la fonction

 return y

- Une boucle for

for i in range(n) :

 corps de la boucle

- Une boucle while
while condition :
 corps de la boucle

La structure conditionnelle

if condition : ou if condition :
 action 1 action 1
else :
 action 2

if condition 1 :
 action 1
elif condition 2 :
 action 2
else :
 action 2

Exercice 1 : (Sur les suites numériques)

Pour tout $n > 0$, on pose $U_n = 1/(n^2)$.

- a) Écrire une fonction qui renvoie le n-ième terme de U_n .
- b) Écrire une fonction qui renvoie les premiers termes de la suite U_n .
- c) Écrire une fonction qui calcule la somme des n premiers termes.
- d) Écrire une fonction qui représente graphiquement les n premiers termes.
(indication : on pourrait utiliser le module matplotlib)

Exercice 2 : (sur les listes)

a) Écrire un programme demandant la saisie de deux nombres, rappelle leurs valeurs puis affiche leur somme.

b) Écrire un programme qui :

- crée une liste vide a
- demande la saisie d'un nombre de départ, d'un nombre d'arrivée et d'un pas
- ajoute à la liste a les trois nombres
- ajoute à la liste a une autre liste, constituée à partir des nombres saisis (nombre de départ, nombre maximal, pas d'augmentation)
- affiche la liste
- affiche les pas derniers éléments de la liste

c)

Initialiser la liste a à la valeur [1,1,1,3,1,4,1,1,3,6,1,2,3,1,4]

afficher a

afficher le nombre de 3 présents dans a

afficher la position du dernier 3

retirer le dernier 3 de la liste

afficher a

afficher le nombre de 3 présents dans a

afficher la position du dernier 3

Exercice 4 (Un test)

Écrire un programme qui :

- demande à l'utilisateur de saisir une chaîne de caractères
- affiche un message signalant que la chaîne contient la lettre 'e' si c'est le cas, et un autre message sinon.

Exercice 5 (une boucle for)

Écrire un programme

1. crée la liste *a* de tous les nombres entiers inférieurs à 100
2. demande à l'utilisateur de saisir un nombre *diviseur*
3. crée à partir de *a* la liste de tous les nombres inférieurs à 100 qui sont multiples de *diviseur*

Par exemple, si l'utilisateur saisit 19, le programme renvoie [0, 19, 38, 57, 76, 95].

Exercice 6 (une boucle while)

Écrire un programme qui :

- crée une liste des nombres entiers compris entre 3 et 15 inclus
- demande à l'utilisateur des nombres et tant que ces nombres sont présents dans la liste, les en retire
- si le nombre n'est pas dans la liste, le programme s'arrête en affichant la liste initiale, puis la liste finalement obtenue.

Exercice 7 : Une fonction !

Définir une fonction prenant comme paramètres l'année *puis* le numéro de mois (par exemple, 3 pour mars) et qui renvoie le nombre de jours du mois correspondant. Les valeurs par défaut sont 2016 pour l'année et 4 pour le mois.

Exercice 8 : Deux fonctions !!

Définir les fonctions suivantes :

a) $f(x) = x^2$

b) $g(x,y) = x^4 + y^4$

Exercice 9 : (Un algorithme de tri par sélection)

a) Écrire une fonction *abmax* renvoyant le plus grand de deux nombres *a* et *b*.

b) Écrire une fonction *vmax* de paramètre d'entrée un vecteur *v* renvoyant le plus grand élément de *v*.

Tester avec *a* = [2, 55, 4, 66, 8, 9].

c) Écrire une fonction `permut` de paramètres entrée un vecteur `v` ainsi que deux entiers `i` et `j` permutant les éléments `v [i]` et `v [j]`, le vecteur initial étant ainsi modifié.

Tester avec `a= [2 ,55 ,4, 66 ,8, 9]`.

Tri par sélection. Soit un vecteur `v` constitué de `n` éléments à trier dans l'ordre décroissant.

Algorithme :

- On cherche l'indice du plus grand élément, soit `i_pg`. On permute `v [i_pg]` et `v[0]`.
- On cherche l'indice du plus grand élément parmi `v [1]`, `v [2]`, ... , `v [n- 1]` , noté encore `i_pg`. On permute `v [i_pg]` et `v [1]`. Etc ...

d) Écrire une fonction `ind_du_max` de paramètres d'entrée un vecteur `v` et un entier `k` renvoyant `i_pg` l'indice du plus grand élément parmi ceux dont l'indice est supérieur ou égal à `k`.

Tester avec `b= [22, 5, 10, 1 , 8, 4]` pour différentes valeurs de `k`.

e) Tester la fonction `triselection` à l'aide du vecteur `c= [0, 1, 2, 3, -8 , - 7]` à trier dans l'ordre décroissant.

Vérifier qu'elle correspond à l'algorithme précédent.

`def triselection (v) :`

`L=len(v)`

`for i in range(L-1) :`

`i_pg=ind_du_max(v, i)`

`permut(v ,i, i_pg)`

`print (i, i_pg, v) # enlever si vecteurs trop longs`

Compléter le programme en affichant les vecteurs en cours de tri et le vecteur final.