Université de Sfax	Année Universitaire : 2016-2017
Institut Préparatoire aux Etudes d'Ingénieurs de Sfax	Section : BG1 Matière : Informatique — Durée : 1h

Devoir de contrôle du 2ème semestre

NB : Répondre sur la feuille de réponses.

Exercice 1 (4 points)

Pour chacune des situations décrites dans le tableau ci-dessous, choisir l'objet Python itérable qui convient parmi la liste suivante : **str**, **tuple**, **list**, **set**, **dict**.

Situation	Itérable	Justification
a) On veut transmettre une collection d'éléments hétérogène et ordonnée à une fonction. Il sera interdit à la fonction de modifier cette collection.		
b) On veut représenter les nombres d'occurrence d'un ensemble de mots dans une chaîne de caractères.		
c) On veut construire une collection d'éléments sans redondance.		
d) On veut transmettre une collection ordonnée d'éléments à une fonction qui va les y ranger de nouveau.		

Exercice 2 (7 points)

1) Soit la fonction Python `qui_suis_je` :

```
def qui_suis_je(a,b=1):
    c=[]
    for d in a:
        e=d+b
        c = c + [e]
    return(c)
```

Compléter le tableau avec les résultats d'exécution correspondant aux instructions données.

2) Pour chacune des fonctions suivantes, trouver l'erreur et corriger-la :

```
def erreur1(liste):
    res = 0
    i = 0
    while i < len(liste):
        res = res + i
    return(res)
```

```
def erreur2(chaine):
    res = []
    for lettre in chaine:
        res = res + lettre
    return(res)
```

```
def erreur3(chaine):
    dico={}
    for lettre in chaine:
        if lettre in dico:
            dico[lettre] == dico[lettre]+1
        else:
            dico[lettre] == 1
    return(dico)
```

Exercice 3 **(9 points)**

Pour faire la compression des nombres entiers, nous allons utiliser des **listes Python** contenant des chiffres et appliquer le principe suivant : chaque séquence d'éléments identiques est représentée par un couple **(V, L)** où **V** est la valeur qui se répète le long de la séquence et **L** est sa longueur.

Exemple :

La compression de la liste **D** suivante :

D =	1	1	1	1	1	1	1	1	0	0	0	0	2	2	1

Retourne une liste **C** représentée par :

C =	1	8	0	4	2	2	1	1

Où on a, pour chaque indice **pair** de **C** :

- Dans la case **C [i]**, la valeur **V** d'une séquence d'entiers identiques dans la liste **D** ;
- Dans la case **C [i+1]**, la longueur **L** de cette séquence.

1) Ecrire une fonction Python ***saisieChiffre*** permettant de saisir un entier **a** compris entre 0 et 9.
2) Ecrire une fonction Python ***saisieListe*** permettant de saisir une liste de taille **n** (la taille est donnée en paramètre). La liste saisie ne contient que des chiffres de 0 à 9.
3) Ecrire une fonction Python ***codage*** permettant de faire la compression d'une liste **D** donnée en paramètre, en appliquant la méthode décrite ci-dessus. Cette fonction retourne une liste **C** ayant subi une compression.
4) Ecrire une fonction Python ***decodage*** permettant de faire la décompression d'une liste **C** donnée en paramètre. Le résultat de cette fonction est une liste **D** étant égale à la liste originale.
5) Ecrire le programme principal qui demande la saisie d'une liste de taille 15, d'afficher les résultats de son **codage** et **décodage**.

Université de Sfax	Année Universitaire : 2016 - 2017
Institut Préparatoire aux Etudes d'Ingénieurs de Sfax	Section : BG1 – Matière : Informatique
Date : 23/02/2017	Durée : 1 Heure
Nom : ..... Prénom :	Section : ..... N° étudiant : .....

Devoir de contrôle du 2ème semestre

Feuille de réponses

Exercice 1

Situation	Itérable	Justification
a)	.....	....
b)	.....	....
c)	.....	....
d)	.....	....

Exercice 2

1) Compléter le tableau suivant :

Exécution	Résultat
`print (qui_suis_je ([2,3,4],10))`	.....
`print(qui_suis_je ("abc","z"))`	.....
`print(qui_suis_je ([1,2,3],"a"))`	.....
`print(qui_suis_je ([1,9,6]))`	.....

NE RIEN INSCRIRE ICI

2) Trouver et corriger les erreurs dans les fonctions Python suivantes :

Fonction Python	Correction d'erreurs

`def erreur1(liste):`
`  res = 0`
`  i = 0`
`  while i < len(liste):`
`  res = res + i`
`  return(res)`
	.....
	.....
	.....

`def erreur2(chaine):`
`  res = []`
`  for lettre in chaine:`
`  res = res + lettre`
`  return(res)`
	.....
	.....
	.....

`def erreur3(chaine):`
`  dico={}`
`  for lettre in chaine:`
`  if lettre in dico:`
`  dico[lettre] == dico[lettre]+1`
`  else:`
`  dico[lettre] == 1`
`  return(dico)`
	.....
	.....
	.....
	.....

Exercice 3 (programmation python)