

Exercices pour pratiquer ITI1520 (Final 2017)

1. Lesquels des programmes suivants contiennent un appel de fonction?

(I) `type(55.5)`

(II) `def maFonction(x):
 return x + 10`

(III) `surface(2, 4)`

(IV) `print("Bonjour")`

- (a) seulement (III)
- (b) seulement (II) et (III)
- (c) seulement (I), (III), et (IV)
- (d) seulement (I), (II), (III), et (IV)
- (e) Aucune des d_eclarations ci-dessus.

2) Si `a` est un nombre entier de trois chiffres, lesquelles des lignes suivantes donnent le chiffre du milieu `a`? (Par exemple, si `a` est 234, le chiffre du milieu est 3.)

(I) `(a // 10) % 10`

(II) `(a % 100) // 10`

(III) `(a % 10) // 10`

- (a) seulement (I)
- (b) seulement (II)
- (c) seulement (III)
- (d) seulement (I) et (II)
- (e) (I), (II), et (III).

3. Qu'est-ce que le programme Python suivant va a_cher sur l'_ecran?

```
def maFonction(x):  
    i=0  
    while i < len(x):  
        print(x[i], end=" ")  
        while (i<len(x)-1 and x[i] == x[i+1] ):  
            i=i+1  
        i=i+1  
#programme principal  
maFonction([3,-4,-4,5,5,5,7,0,0,0,6])
```

4. Que fait la fonction Python suivante?

```
def mafonction(a):  
    """(list of int) -> None  
    Precondition: len(a) >= 3"""  
    while len(a) > 3:  
        a.remove(min(a)) #la fonction Python min retourne le minimum dans une liste
```

la fonction Python *remove* élimine la première occurrence d'une valeur dans la liste

- (a) Modifie la liste pour éliminer les len(a)-4 plus petites valeurs de la liste a.
- (b) Modifie la liste a pour retenir seulement les 3 valeurs plus grandes.
- (c) Modifie la liste a pour retenir seulement les 3 valeurs plus petites.
- (d) Modifie la liste a pour retenir seulement les 4 valeurs plus grandes.
- (e) Aucune des d_eclarations ci-dessus.

5. Lesquelles des d_eclarations suivantes sont correctes au sujet de la fonction orange (qui prend comme paramètre un entier positif)?

```
def orange(num):  
    """ (int)->int  
    Precondition: num est positif """  
    t = 0  
    for i in range(num):  
        t = t + 10  
    return t
```

- (a) La fonction calcule la valeur de 10 plus num.
- (b) La fonction calcule the valeur 10 à la puissance num.
- (c) La fonction calcule 10 multiplié par num.
- (d) La fonction calcule le factoriel de num
- (e) Aucune des déclarations ci-dessus

6.

```
if heure >=0 and heure <=12:  
    return False  
else:  
    return True
```

Lesquelles des déclarations suivantes sont équivalentes au code ci-dessus?

- I) return heure >=0 and heure <=12
- II) return not(heure >=0 and heure<=12)
- III) return heure <0 or heure >12

- (a) I seulement
- (b) II seulement
- (c) III seulement
- (d) I et III
- (e) II et III

7. Voici une fonction qui trie une liste a (du plus grand au plus petit):

```
0: def bubble_sort(a):
1:     for i in range(len(a)):
2:         for j in range(len(a)-1):
3:             if a[j] < a[j+1]:
4:                 a[j], a[j+1] = a[j+1], a[j]
```

Si la liste a contient 10 000 éléments, la ligne 3 de bubble_sort(a) s'exécute

- (a) 9 900 fois
- (b) 10 000 fois
- (c) 99 900 fois
- (d) 100 000 fois
- (e) 99 990 000 fois

8. Si la liste a contient 10 000 éléments qui sont déjà triés du plus grand au plus petit, la ligne 4 de bubble_sort(a) s'exécute :

- (a) 0 fois
- (b) 10 000 fois
- (c) 9 900 fois
- (d) 99 900 fois
- (e) 99 990 000 fois

9. Considérez la fonction suivante:

```
def mafonction(n):
    """(int)->None
    Precondition: n est un entier positif
    """
    for i in range(1,n+1):
        if(n%i == 0):
            print(i, end=" ")
    print()
```

Que fait la fonction?

- (a) affiche tous les entiers premiers inférieurs ou égaux à n
- (b) affiche tous les entiers non premiers inférieurs ou égaux à n
- (c) affiche tous les entiers divisibles par i
- (d) affiche tous les diviseurs de n+1
- (e) affiche tous les diviseurs de n

10. Le corps de la fonction suivante manque.

```
def meme_chaine(s1,s2):  
    """(str,str)->bool  
    Precondition: len(s1)==len(s2) > 1  
    Pour deux chaines de caracteres s1 et s2 de meme taille, la fonction retourne True  
    si les deux chaines contiennent les memes caracteres dans la meme orde et False  
    sinon.  
    """
```

Quel code doit-on y ajouter pour que la description soit correcte?

(I)

```
for i in range(len(s1)):  
    for j in range(len(s2)):  
        if s1[i] != s2[j]:  
            return False  
    return True
```

(II)

```
for i in range(len(s1)):  
    if s1[i]==s2[i]:  
        return True  
return False
```

(III)

```
for i in range(len(s1)):  
    if s1[i]!=s2[i]:  
        return False  
return True
```

- (a) I seulement
- (b) II seulement
- (c) III seulement
- (d) I et II
- (e) I et III

11. Une partie de la fonction suivante manque.

```
def common_chars(s1, s2):  
    """(str, str) -> str  
    Retourne une nouvelle chaîne avec les caractères de s1 qui arrivent au moins une  
    fois en s2. Les caractères dans le résultat conservent leur ordre d'occurrence en s1.  
  
    >>> common_chars('abc', 'ad')  
    'a'  
    >>> common_chars('a', 'a')  
    'a'  
    >>> common_chars('abb', 'ab')  
    'abb'  
    >>> common_chars('abracadabra', 'ra')  
    'araaara'  
    """  
  
    res = "  
    # code manque  
    return res
```

Quel est le fragment de code qui manque?

- (a)

```
for ch in s1:  
    for ch in s2:  
        res = res + ch
```
- (b)

```
for ch in s2:  
    if ch in s1:  
        res = res + ch
```
- (c)

```
for ch in s1:  
    if ch in s2:  
        res = res + ch
```
- (d)

```
if ch in s2:  
    for ch in s1:  
        res = res + ch
```

12. Quelle est la description correcte de la fonction suivante?

```
def mafonction(n):  
    """ (int)->(int)  
    Precondition: n>=1 """  
    for i in range(n):  
        total=1  
        total=total*n
```

```
return total
```

- (a) elle retourne n^n
- (b) elle retourne $n*n$
- (c) elle retourne n
- (d) elle retourne 1
- (e) elle donne erreur pour $n \geq 1$

13. Qu'affiche le programme suivant ?

```
def maFonction(x, n):  
    if n > 0:  
        print(x[n - 1], end = " ")  
        maFonction(x, n-1)
```

```
a = [10, 20, 30, 40, 50]  
maFonction(a, len(a))
```

- (a) 10 20 30 40 50.
- (b) 10 20 30 40 50 et donne un message d'erreur (index out of range).
- (c) 50 40 30 20 10.
- (d) 50 40 30 20 10 et donne un message d'erreur (index out of range).
- (e) Aucune des d_éclarations ci-dessus.

14. Quelle est la description correcte de la fonction suivante?

```
def pomme(L, c):  
    """ (list, str) -> int  
    Precondition: L est une liste de chaines de caracteres.  
    Chaque element de L contient au moins un caractere.  
    len(c) == 1, i.e., c contient un seul caractere  
    """  
    res = []  
    for elem in L:  
        if elem[0] == c:  
            res.append(elem)  
    return len(res)
```

- (a) Retourne la liste de chaines de caractères qui commence avec x .
- (b) Retourne le nombre total de caractères de toutes les chaines de L .
- (c) Retourne le nombre de chaines de caractères de L qui contiennent x .
- (d) Retourne le nombre de chaines de caractères qui commencent avec x .
- (e) Aucune des déclarations ci-dessus.

15. Lesquelles des fonctions suivantes calculent correctement et retournent la somme $1/2 + 2/3 + 3/4 + \dots + 99/100$?

```
def f1():
    somme = 0
    for i in range(1, 99):
        somme = somme + i / (i + 1)
    return somme

def f2():
    somme = 0
    for i in range(1, 100):
        somme = somme + i / (i + 1)
    return somme

def f3():
    somme = 0
    for i in range(99):
        somme = somme + (i+1) / (i + 2)
    return somme
```

- (a) f1 seulement
- (b) f2 seulement
- (c) f3 seulement
- (d) f1 et f3
- (e) f2 et f3

16. Considérez le programme suivant:

```
def contient(val, lst):
    """(object, list) -> bool
    Retourne True si val est un element d'une des listes de la liste 2D lst.
    >>> contient('moogah', [[70, 'bleu'], [1.24, 90, 'moogah'], [80, 100]])
    True
    """
    trouve = False # On n'a pas encore trouve val dans la liste.
    # CODE MANQUE
    return trouve
```

Lesquels des fragments de code suivants sont corrects pour compléter cette fonction?

I

```
for sousListe in lst:
    if val in sousListe:
        trouve = True
```

II

```
for i in range(len(lst)):
    for j in range(len(lst[i])):
        if lst[i][j] == val:
            trouve = True
```

III

```
for i in lst:
    if i == val:
        trouve = True
```

- (a) I seulement
- (b) II seulement
- (c) III seulement
- (d) I et II
- (e) II et III

17. Quelle est la description correcte de la fonction suivante, si L est une liste des entiers qui n'est pas vide?

```
def gallerie(L):
    X=sorted(L) # sorted(L) retourne une version trie de la liste L
    if len(X)==1:
        return True
    for i in range(len(X)):
        if i==0 and X[i]!=X[i+1]: # X[i] est le premier element
            return True
        elif i==len(L)-1 and X[i]!=X[i-1]: # X[i] est le dernier element
            return True
        elif i!=0 and i!=len(L)-1: # pas le premier ou le dernier
            if X[i]!=X[i-1] and X[i]!=X[i+1]:
                return True
    return False
```

- (a) Retourne True si les éléments de L sont tous différents (i.e., si L ne contient pas de doubles), et False sinon.
- (b) Retourne False si les éléments de L sont tous différents (i.e., si L ne contient pas de doubles), et True sinon.
- (c) Retourne True si L contient au moins un élément qui arrive exactement une fois dans L, et False sinon.
- (d) Retourne False si L contient au moins un élément qui arrive exactement une fois dans L, et False sinon.
- (e) Aucune des déclarations ci-dessus.

18. Quelle est la description correcte de la fonction suivante, si L est une liste des entiers triés en ordre croissant qui n'est pas vide?

```
def maFonction ( x , deb , fin , L ) :  
    if deb>fin : return " fini "  
    t =(deb+fin)//2  
    if x==L[t] : return t  
    if x<L[t] : return maFonction(x ,deb,t-1,L)  
    else : return maFonction( x, t +1 , fin ,L)
```

- (a) Retourne la plus petite valeur de la liste L.
- (b) Retourne la plus grande valeur de la liste L.
- (c) Vérifie si la valeur x se trouve dans la liste L et retourne sa position.
- (d) Vérifie si la valeur x se trouve dans la liste L et retourne sa valeur.
- (e) Aucune des réponses ci-dessus.

19. Imaginez que la liste L dans la question précédente contient 1000 éléments triés en ordre croissant (du plus petit au plus grand), combien va-t-on inspecter d'éléments de L pour l'algorithme de recherche de la question 18 ?

- (a) au plus 1
- (b) au plus 5
- (c) au plus 10
- (d) au moins 100
- (e) au moins 1000

20. Imaginez que la liste L dans la question précédente 18 contient 1000 éléments qui ne sont pas triés, combien va-t-on inspecter d'éléments de L pour faire la même recherche que l'algorithme de recherche de la question 18 et ceci de la façon la plus efficace possible ?

- (a) au plus 1
- (b) au plus 5
- (c) au plus 10
- (d) au plus 100
- (e) au plus 1000

21. Considérant la classe suivante :

```
class Domino () :  
    def __init__(self, pa, pb) :  
        self.pa, self.pb = pa, pb  
    def affiche_points (self) :  
        print ("Face A :",self.pa,end=' ')  
        print ("Face B :",self.pb)  
    def valeur (self) :  
        return (self.pa + self.pb)
```

Qu'affiche le programme suivant :

```
d1 = Domino (2,6)  
d2 = Domino (4,3)  
print ("Total des points : ",d1.valeur () + d2.valeur ())
```

- (a) Total des points : 15
- (b) Total des points : 6
- (c) Total des points : 9
- (d) Erreur
- (e) Aucune des réponses ci-dessus.

22. Qu'affiche le programme suivant :

```
class JeuDeCartes() :  
    couleur = ("Pique","Trèfle","Coeur","Carreau")  
    valeur = (2,3,4,5,6,7,8,9,10,'valet','dame','roi','as')  
    def __init__(self) :  
        #~ Construction de la liste des 52 cartes  
        self.carte=[]  
        for coul in range (4) :  
            for val in range (13) :  
                self.carte.append ((val, coul))  
        return
```

```
jeu = JeuDeCartes()  
print(Jeu.couleur)
```

- (a) ('Pique', 'Trèfle', 'Coeur', 'Carreau')
- (b) ('Pique')
- (c) (2,3,4,5,6,7,8,9,10,'valet','dame','roi','as')
- (d) Message d'erreur

23. Considérant la même classe précédente. Qu'afficherait le programme suivant :

```
print(JeuDeCartes.couleur)
```

- (a) ('Pique', 'Trèfle', 'Coeur', 'Carreau')
- (b) ('Pique')
- (c) (2,3,4,5,6,7,8,9,10,'valet','dame','roi','as')
- (d) Message d'erreur