

IT11520, Exemples des questions d'examen partiel

1. Quelle est la valeur de l'expression `not 16 > 3 * 2 and 10 == 5 * 2`?  
(a) 6                    (c) True                (e) Aucune de  
(b) 2                    (d) False                ci-dessus
2. Si `a` et `b` sont de type `int`, l'expression `a // b` a le type  
(a) `float`            (b) `int`                (c) `str`                (d) `bool`                (e) `object`
3. Lesquels des programmes suivants I, II, et III affichent les memes valeurs que ce programme:

```
for val in range(2,6):  
    print(val)
```

I

```
i = 2  
while (i <= 6):  
    print(i)  
    i = i + 1
```

II

```
i = 2  
while (i < 6):  
    i = i + 1  
    print(i)
```

III

```
i = 2  
while (i < 6):  
    print(i)  
    i = i + 1
```

- (a) I only  
(b) II only  
(c) III only  
(d) II and III  
(e) I, II, and III
4. Laquelle est la description la plus correcte de la fonction suivante (qui prend comme parametre un entier positif)?

```

def orange(num):
    ''' (int)->int
    Precondition: num est positif'''
    t=0
    for i in range(1, num, 2):
        print(i)
        t = t + i
    return t

```

- (a) La fonction calcule la somme des entiers de 2 al num.
- (b) La fonction calcule la somme des entiers de 1 a num-1.
- (c) La fonction calcule la somme des entiers 1 a num.
- (d) La fonction calcule la somme des entiers impaires de 1 a num
- (e) aucune de ci-dessus

5. Dans le programme suivant

```

m = 3
n = 8
x = [m + n, m % n]

```

la valeur de x est:

- (a) [11, 3]
- (b) [11, 0]
- (c) 14
- (d) 11
- (e) aucune de ci-dessus

6. Quel est le type de la variable x?

```

def foo(a, b):
    print(a + b)

```

```

x = foo(4,8)

```

- (a) str
- (b) int
- (c) float
- (d) NoneType

7. Considérez le programme suivant avec une partie de code qui manque.

```
>>> a = [1,2,3]
>>> b = a
>>> #missing code
>>> print(a,b)
[1, 'A', 3] [1, 'A', 3]
```

Le code qui manque devrait etre:

- (a) `b[-2] = 'A'`
- (b) `b = [1, 'A', 3]`
- (c) `a[1] = 'A'`
- (d) None of the above

8. Quel est le type pour les parametres et le resultat dans la definition de la fonction suivante?

```
def test(w1, w2):
    '''
    Retourne True si w1 existe en w2.

    >>> test('el', 'hello')
    True
    '''
```

- (a) `(str,str)->str`
- (b) `(str,str,str)->bool`
- (c) `(str)->bool`
- (d) `(str,str)->bool`

9. Considerez le programme suivant:

```
def kiwi(x):
    res = 0
    for i in range(x):
        res = res + x
    return res
```

Qu'est ce qui se passe si on deplace l'initialisation de la variable `res` (la ligne `res = 0`) a l'interieur de la boucle, comme la premiere ligne dans la boucle?

- (a) La fonction va retourner `x` au lieu de `x * x`
- (b) La fonction va donner une erreur
- (c) La fonction va retourner `x * x`
- (d) La fonction va retourner `0` au lieu de `x * x`

10. Une partie de la fonction suivante manque.

```
def deplace_a_la_gauche(x):
    '''
    (list) -> NoneType
    Deplace les elements de la liste a la gauche. La valeur du premier
    element est copie dans le dernier element.
    >>>x = [1,2,3,4]
    >>>shift_right(x)
    [2,3,4,1]
    '''
    temp = x[0]

    # code manque

    x[len(x)-1] = temp
```

La partie de code qui manque devrait etre:

- (a) 

```
for i in range(len(x)):
    x[i] = x[i+1]
```
- (b) 

```
for i in range(len(x)-1):
    x[i+1] = x[i]
```
- (c) 

```
for i in range(1, len(x)-1):
    x[len(x)-i] = x[len(x)-i-1]
```
- (d) 

```
for i in range(len(x)-1):
    x[i] = x[i+1]
```