

DS 2

Informatique pour tous, première année

Julien REICHERT

Exercice 1 : QCM.

- Combien de nombres peut-on représenter en binaire sur 32 bits ?
 - $2^{31} - 1$
 - 2^{31}
 - 2^{32}
 - $2^{32} - 1$
- La représentation en « complément à deux » (ou « binaire signé ») sur 16 bits du nombre 50000...
 - commence par un 0
 - commence par un 1
 - est en réalité celle d'un autre nombre
 - est la même que celle de -50000 à un bit près
 - termine par 1
 - termine par 0
 - est 1100001101010000
 - est 0011110010101110
- Qu'est-ce qui apparaît dans la console lorsqu'on y écrit `[1, 2, 3, 4, 5, 6, 7, 8, 9][-3]` ?
 - 123456786
 - [1, 2, 4, 5, 6, 7, 8, 9]
 - 6
 - 7
 - Une erreur de syntaxe
 - Une erreur d'indexation
- Parmi ces expressions, lesquelles ont une longueur de sept (au sens de la fonction `len`) ?
 - 1234567
 - "abcdefg"
 - ["1234567"]
 - [1,2,3,4,5,6,7]
 - list("abcdefg")
 - range(1,7)
- Parmi ces variables, lesquelles ont une longueur de dix (idem) après exécution du script ci-dessous ?
 - l1
 - l2
 - l3
 - l4
 - l5
 - l6

```
l1 = []
i1 = 1
while i1 <= 4:
    l1.append(i1*[i1])
    i1 = i1 + 1
```

```
l2 = []
i2 = 1
while i2 <= 4:
    l2.extend(i2*[i2])
    i2 = i2 + 1
```

```
l3 = []
for i3 in range(4):
    for j in range(i3 % 2 + 2):
        l3.append([42])
```

```
14 = list(range(42))[2:22:2]
```

```
15 = [0] + 14[: -1]
```

```
16 = 15[1:]
```

Exercice 2 : On considère les scripts Python (a), (b), (c), (d), (e) et (f). Déterminer lesquels provoquent une erreur lors de l'évaluation du script ou de l'appel de la fonction et expliquer les erreurs. Déterminer également ce qui apparaît dans la console lors de l'évaluation des scripts.

(a)

```
def f1(n):  
    i = 0  
    while i < n:  
        print(i)  
        i = i + 1
```

```
f1(42)
```

(b)

```
def f2(a):  
    n = len(a)  
    for i in range(n-1, 0):  
        print(a[i])
```

```
f2([1, 2, 3])
```

(c)

```
def f3(n):  
    for i in range(n):  
        print(i)  
    return n
```

```
f3(42)
```

(d)

```
def f4(l):  
    for i in range(l):  
        print("l[i] vaut :")  
        print(l[i])  
        print("\n")  
    return i
```

```
f4([1, 2, 3])
```

(e)

```
def f5(n):  
    while n > 0:  
        print(n)  
        n - 1  
        print("C'est fini !")  
    return n
```

```
f5(42)
```

(f)

```
def f6(n):  
    for i in range(n):  
        return i  
    if n % 2 == 0:  
        print(1/0)  
    else:  
        print("Ok")
```

```
f6(42)
```

Exercice 3 : Écrire une fonction qui prend en entrée une liste de nombres et qui renvoie le minimum et le maximum de cette liste. On utilisera une boucle bien choisie.

Exercice 4 : Écrire une fonction qui prend en entrée une liste de nombres et qui renvoie le nombre d'éléments pairs de cette liste. On utilisera une boucle bien choisie.