

QCM

TEST

**Contrôle Continu
Algo - Prog L1
Examen du 25/10/2024**

**Durée : 1h 30 minutes (2h
Tiers-Temps)**

Nom :

Prénom :

Numéro d'étudiant :

- L'usage de la calculatrice, de tout appareil électronique (dont smartphones et montres connectées) est interdit et sera assimilé à un cas de triche.
- document autorisé : feuille manuscrite A4 recto-verso, avec votre nom et prénom clairement indiqués.
- Les questions 1 à 11 ont **une seule** bonne réponse. La question 12 demande du code.
- Chaque question de 1 à 11 a un barème sur n points. Une bonne réponse donne n points, une mauvaise réponse compte $-n/2$ points (négatifs), une réponse multiple ou une absence de réponse donnent 0 point. La question 12 a un barème dont la note est laissée au correcteur.

Question 1 (1 point) Que vaut $5 + 4 * 7 // 5 \% 6 - 3 + 2 * 5 - 1$

☐ 11☐ 9☒ 16

Question 2 (1 point) Que vaut $(-13 // 3) * (-13 \% 3)$?

☐ 4☒ -10☐ une autre réponse

Question 3 (1 point) Que vaut `not(4 != 23 % 5 and not(3 % 2 != 3 // 2))` ?

☐ True☒ False

Question 4 (1,5 point) `mot="Algorithmes de recherche"`; Que vaut `mot[1:13:3] + mot[-5:-2]` ?

☐ "rhserc"☐ "rhseerc"☐ "lrhseerc"☒ "lrhserc"

Question 5 (1,5 point) Que vaut $5 \% 2 + 3 // 2 * 3 + \text{int}('4' < '25')$?

Aide : `int(False)` vaut 0 et `int(True)` vaut 1

☐ 5☐ 2☐ 1☒ 4

Question 6 (2 points) Voici un code python. Quel est l'affichage final produit ?

```
ch = 'a b c d e f g h'
for i in range(2, 4):
    ch = ch[-2*i-2:-1] + ch[i:2*i+1]
print(ch)
```

☐ cdefgcde☐ cdefgcd☐ cdefgcfgc☒ cdefgcdfgcd☐ defgcfgc☐ une erreur est signalée

Correction

Question 7 (2 points) Voici un code python. Quel est l'affichage final produit ?

```
n = 60
c = ''
while n >= 1 :
    c = str(n % 3) + c
    n = n // 3

print("c=" + c)
```

☐ c=2022

☐ c=2021

☒ c=2020

☐ c=2023

Question 8 (2 points) Voici un code python. Quel est l'affichage final produit ?

```
res = []
t = (5, 1, 4, 0, 3, 2)
for i in range(len(t)) :
    j = t[-1-i]
    res.append(t[j])
print(res)
```

☐ [5, 1, 4, 0, 3, 2]

☐ [0, 5, 4, 1, 3, 2]

☐ [2, 3, 4, 5, 0, 1]

☒ [4, 0, 5, 3, 1, 2]

Question 9 (1,5 point) Voici un code python. Combien affiche-t-il de caractères '*' ?

```
for i in range(5) :
    for j in range(2, 11, 3) :
        if i + j > 6 :
            print('*', end='')
```

☐ 5

☐ 6

☒ 8

☐ 4

Question 10 (1 point) Voici un code python. Quel est l'affichage produit ?

```
mot = 'aBCdE'
for i in range(len(mot)) :
    if mot[i].islower():
        mot[i] = 'x'
    else:
        mot[i] = 'X'
print(mot)
```

☒ une erreur est signalée

☐ xXXxX

☐ AbcDe

☐ abcde

Question 11 (1,5 point) Voici un code python. Quel est l'affichage produit ?

```
def f(liste) :
    liste.append(liste[-1]+liste[-2])

maListe = [1, 1]
n = 4
for i in range(n) :
    f(maListe)

print(maListe[-1])
```

☐ 5

☒ 8

☐ 3

☐ 13

Question 12 (4 points) Voici deux fonctions python dont le code est à écrire :

- la fonction **telDuPlusVieux** retourne le numéro de téléphone de la personne la plus âgée de la liste passée en paramètre. Il est bien entendu que le code permettra de retourner le bon résultat, même si la liste passée en paramètre évolue !
- la fonction **nomsDesPersonnesDontLeTelCommencePar** retourne la liste composée des noms des personnes de la liste passée en paramètre et dont le numéro de téléphone commence par le paramètre nn (par exemple, nn pourra valoir '06'). Il est bien entendu que le code permettra de retourner le bon résultat, même si la liste passée en paramètre évolue !

```
def telDuPlusVieux(liste):
    age_max = liste[0]["age"]
    tel_max = liste[0]["tel"]
    for personne in liste :
        if personne["age"] > age_max :
            age_max = personne["age"]
            tel_max = personne["tel"]
    return tel_max

def nomsDesPersonnesDontLeTelCommencePar (liste, nn) :
    resultat = []
    n = len(nn)
    for personne in liste :
        if personne["tel"][0:n] == nn :
            resultat.append(personne["nom"])
    return resultat

o221 = {"nom" : "Dupont", "tel" : "06.24.65.31.20" "age" : 23}
o432 = {"nom" : "Durant", "tel" : "06.31.29.18.14" "age" : 24}
o117 = {"nom" : "Dupuis", "tel" : "07.35.41.68.71" "age" : 32}
o141 = {"nom" : "Dumont", "tel" : "07.11.55.41.23" "age" : 29}

liste = [o221, o432, o117, o141]

print (telDuPlusVieux(liste))
print (nomsDesPersonnesDontLeTelCommencePar (liste, '06'))
```