

UNIVERSITE DE LAUSANNE
Faculté des Hautes Etudes Commerciales

Professeurs	Matière	Session
Benoît Garbinato	Algorithmes et Pensée Computationnelle	Winter 2018

Durée : 1 heure.

Aucun document n'est autorisé. L'usage de la calculatrice ou de tout autre matériel électronique est interdit.

Chaque question ne peut avoir qu'une seule bonne réponse.

A la fin de l'examen, vous rendrez cet énoncé et la feuille de réponses distribuée avec l'énoncé. Les deux documents doivent porter votre nom, prénom et numéro d'étudiant.

<p><i>Nom et prénom:</i></p> <p>.....</p> <p><i>No matricule:</i></p> <p>.....</p> <p><i>No de place:</i></p> <p>.....</p>
--

Question 1 [Q01]

Soit le code suivant en Python 3.5, quelle affirmation est correcte ?

```

1 mountains = ["jungfrau", "eiger", "moench"]
2 for summit in mountains:
3     print("I will climb to the summit of the {}".format(summit))
4
5 i = 0
6 while i < len(mountains):
7     print("I will climb to the summit of the {}".format(mountains[i]))
8     i = i + 2

```

- A Les lignes suivantes sont affichées dans la console standard après son exécution:
 - I will climb to the summit of the jungfrau
 - I will climb to the summit of the eiger
 - I will climb to the summit of the moench
 - I will climb to the summit of the jungfrau
 - I will climb to the summit of the moench
- B La première boucle n'affiche pas le contenu entier de moutains.
- C L'affichage de la seconde boucle devient identique à la première si on remplace $i = i + 2$ par $i = i + 3$
- D La seconde boucle affiche le contenu entier de moutains.

Question 2 [Q02]

Soit le code en Python 3.5 ci-dessous, qu'est-ce qui sera affiché dans la console standard après son exécution ?

```
1 list_nums = [1,1,2,3,3,3,4,4,4,4]
2 print(len(list_nums))
```

- 10
- 4
- 11
- 9

Question 3 [Q03]

Soit le programme ci-dessous en Python 3.5, veuillez sélectionner la ligne de code qui permettra de l'exécuter correctement (à la place des « # » de la ligne n°3).

Description du programme:

Le programme suivant doit afficher tous les nombres entre 2000 et 3200 (les deux inclus) qui sont divisibles par 7 et qui ne sont pas des multiples de 5.

Code du programme:

```
1 l=[]
2 for i in range(2000, 3201):
3     #####Line 3#####
4     l.append(str(i))
5 print(l)
```

- (A) if (i%7 != 0) and (i%5 == 0):
- (B) if (i%7 == 0) and (i%5 == 0):
- (C) if (i%7 != 0) and (i%5 != 0):
- (D) if (i%7 == 0) and (i%5 != 0):

Question 4 [Q04]

Parmi les affirmations suivantes, quelle est la séquence qui permet de correctement exécuter une instruction par le processeur ?

- (A) Instruction Fetch -> Execute -> Decode -> Store
- (B) Instruction Fetch -> Decode -> Store -> Execute
- (C) Instruction Fetch -> Decode -> Store -> Execute -> Write Back
- (D) Instruction Fetch -> Decode -> Execute -> Store

Question 5 [Q05]

Soit le code suivant en Python 3.5, qu'est-ce qui sera affiché dans la console standard ?

```
1 x = 0
2 while x < 4:
3     x = x + 1
4
5 print("x is", x)
```

- (A) x is 0
- (B) x is 3
- (C) x is 4
- (D) x is 5

Question 6 [Q06]

Parmi les lignes de code suivantes en Scala, quel est le code qui permet de convertir la chaîne de caractères (string) `pi_string = '3.1416'` en un type float ?

- A `pi_float = pi_string.toFloat`
- B `pi_float = float(pi_string)`
- C `pi_float = Float(pi_string)`
- D `pi_float = flt(pi_string)`

Question 7 [Q07]

Soit la fonction en Swift: `func checkExerciseRoomAccess(firstYearBachelor: Bool)`. Qu'est-ce qui sera affiché dans la console standard après son exécution si le paramètre `firstYearBachelor` est égal à `true` ?

```

1 func checkExerciseRoomAccess(firstYearBachelor: Bool){
2     var beginner = false
3     if firstYearBachelor {beginner = true}
4     let books: [String] = ["Python for beginners", "Scala in a nutshell", "Swift
5         Programming"]
6     var cpt = 0
7
8     for book in books{
9         if book.contains("Python") || book.contains("Scala") || book.contains("Swift
10            ") {
11             cpt += 1
12         }
13     }
14
15     if beginner && cpt == 3 {
16         print("You can enter in the exercise room :)")
17     }
18
19     else{
20         print("You cannot enter in the exercise room :(")
21     }
22 }

```

- A You can enter in the exercise room :)
- B You cannot enter in the exercise room :(
- C fatal error: Index out of range
- D A et B sont affichées

Question 8 [Q08]

Soit les lignes de code en Swift ci-dessous, quelle expression suivante est égale à `true` ?

```

1 var a = true
2 var b = true
3 var c = !(a && b)
4 var d = a || !(b)

```

- A `c == d`
- B `c == !(d)`
- C `b == d`
- D `a == d`

Question 9 [Q09]

Soit le programme ci-dessous en Scala, qu'est-ce qu'il sera affiché dans la console standard après son exécution ?

```

1 object name extends App{
2     var x = 30
3     var y = 10
4
5     if(x < 30){
6         if(y >= 10){
7             println("tip")
8         }
9     }
10
11    else{
12        println("tip-top")
13    }
14
15    println("top-tip")
16 }
17 }
```

- tip-top
top-tip
- tip
tip-top
- tip
top-tip
- tip
tip-top
top-tip

Question 10 [Q10]

Parmi les affirmations suivantes à propos des fonctions, laquelle est correcte ?

- Une fonction est une séquence d'instructions réalisant une tâche spécifique.
- Il est impossible de passer un paramètre à une fonction.
- Une fonction ne retourne jamais de valeur.
- Aucune des autres propositions mentionnées.

Question 11 [Q11]

Parmi les lignes de code suivantes en Swift, laquelle permet de déclarer une variable ?

- var name = "Alice"
- val age = 2
- let name = "Alice"
- age = 12

Question 12 [Q12]

Soit la fonction Swift: `func getUserAge(name: String) -> Int`. Quel est le type du paramètre et le type de la valeur retournée par cette fonction ?

- Paramètre: Int / Valeur retournée: String
- Paramètre: String / Valeur retournée: String
- Paramètre: Int / Valeur retournée: Int
- Paramètre: String / Valeur retournée: Int

Question 13 [Q13]

Soit le code suivant en Python 3.5:

```
1 mountains = ["jungfrau", "eiger", "moench"]
2 i = 0
3 while i < len(mountains):
4     print("I will climb to the summit of the {}".format(mountains[i]))
5     i = i + 2
```

Nous souhaitons le traduire en Swift, quel code choisissez-vous parmi les options données ci-dessous ?

- (A) let mountains: [String] = ["jungfrau", "eiger", "moench"]
let i = 0
while i < mountains.count {
 print("I will climb to the summit of the \(mountains[i])")
 i = i + 2
}
- (B) let mountains: [String] = ["jungfrau", "eiger", "moench"]
var i = 0
while i < mountains.count {
 print("I will climb to the summit of the " + mountains[i])
 i = i + 1
}
- (C) let mountains: [String] = ["jungfrau", "eiger", "moench"]
var i = 0
while i < mountains.size {
 print("I will climb to the summit of the \(mountains[i])")
 i = i + 2
}
- (D) let mountains: [String] = ["jungfrau", "eiger", "moench"]
var i = 0
while i < mountains.count {
 print("I will climb to the summit of the \(mountains[i])")
 i = i + 2
}

Question 14 [Q14]

Soit la fonction suivante en Python 3.5 qui permet d'afficher la somme des nombres compris entre 1 et une limite (inclue) indiquée en paramètre. Quel est son équivalent récursif parmi les propositions ci-dessous ?

```
1 def exercise(limit) :  
2     sum = 0  
3     for i in range(1,limit+1):  
4         sum = sum + i  
5     return sum
```

- (A) def exercise(limit) :
 if limit == 1:
 return 1
 return limit + exercise(limit-2)
- (B) def exercise(limit) :
 return limit + exercise(limit-1)
- (C) def exercise(limit) :
 if limit == 1:
 return 1
 return limit + exercise(limit-1)
- (D) Aucune des autres propositions mentionnées.

Question 15 [Q15]

Soit la fonction suivante en Python 3.5, quel est l'ensemble des valeurs qui peuvent être retournées par la fonction suivante ? Veuillez uniquement considérer les cas où le paramètre value ne prend que des valeurs entières.

```
1 def exercise(value) :  
2     if value > 0:  
3         return value % 10  
4     elif value == 0:  
5         return value  
6     else:  
7         return -1
```

- (A) [-1, 10]
- (B) [-1, 9]
- (C) [-2, 10[
- (D) [-1, 11[



Feuille de réponses

Exemple: n° 12031123

- ① ① ● ① ① ① ① ①
- ① ① ① ● ● ① ①
- ② ● ② ② ② ② ● ②
- ③ ③ ③ ● ③ ③ ③ ●

← encodez votre numéro d'étudiant ci-contre, et inscrivez votre nom, prénom et numéro de place ci-dessous.

- ① ① ① ① ① ① ① ①
- ① ① ① ① ① ① ① ①
- ② ② ② ② ② ② ② ②
- ③ ③ ③ ③ ③ ③ ③ ③
- ④ ④ ④ ④ ④ ④ ④ ④
- ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤
- ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥ ⑥
- ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦ ⑦
- ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧ ⑧
- ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨ ⑨

Nom:

Prénom:

Place:

Les réponses aux questions sont à donner exclusivement sur cette feuille : les réponses données sur d'autres feuilles ne seront pas prises en compte.

Coloriez complètement et proprement chaque case choisie, afin d'obtenir le meilleur contraste possible par rapport à une case vide. Ne coloriez pas les cases au crayon à papier, n'utilisez pas de couleur rouge (le programme correcteur ne voit pas les couleurs). Utilisez plutôt du noir ou du bleu foncé. Pour corriger une éventuelle erreur de saisie, veuillez utiliser si possible un correcteur liquide (Tipp-Ex).

- QUESTION 1 : ● (B) (C) (D)
- QUESTION 2 : ● (B) (C) (D)
- QUESTION 3 : (A) (B) (C) ●
- QUESTION 4 : (A) (B) (C) ●
- QUESTION 5 : (A) (B) ● (D)
- QUESTION 6 : ● (B) (C) (D)
- QUESTION 7 : ● (B) (C) (D)
- QUESTION 8 : (A) (B) ● (D)

- QUESTION 9 : ● (B) (C) (D)
- QUESTION 10 : ● (B) (C) (D)
- QUESTION 11 : ● (B) (C) (D)
- QUESTION 12 : (A) (B) (C) ●
- QUESTION 13 : (A) (B) (C) ●
- QUESTION 14 : (A) (B) ● (D)
- QUESTION 15 : (A) ● (C) (D)