



EPFL

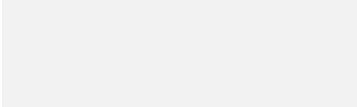
n/a

Ens. : O. Lévêque, M. Stojilovic
Information, Calcul, Communication - (n/a)
Lundi 3 juillet 2023
Durée : 180 minutes

n/a













SCIPER : **999999**

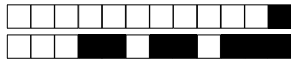
Salle : BLANK

Signature : 

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 16 pages, les dernières pouvant être vides. Ne pas dégrafer.

- Posez votre carte d'étudiant sur la table.
- Document autorisé pour cet examen : partie programmation : tous vos documents de cours ; partie théorique : un formulaire constitué de deux pages A4 recto-verso, manuscrites (ou préparées avec stylet+tablette).
- L'utilisation de tout appareil électronique (calculatrice, ordinateur, smartphone/watch, tablette) est interdite pendant l'épreuve.
- L'examen est composé de trois parties :
 - une partie avec 11 questions à choix multiple sur la théorie ; chaque question admet une seule réponse correcte parmi 4 possibilités : la réponse correcte vaut 2 points ; toute autre option (pas de réponse, réponse fausse, ou plusieurs cases cochées) vaut 0 point. Cette partie vaut donc en tout **22 points**.
 - une partie avec 10 questions sur la programmation ; chaque réponse vaut 2,5 points, et cette partie vaut donc en tout **25 points**.
 - une partie avec 4 exercices sur la théorie, valant chacun 7 points, donc en tout **28 points**.
- Merci d'avance de soigner la présentation de vos réponses !
- Si une question est erronée, les enseignants se réservent le droit de l'annuler.

Respectez les consignes suivantes Read these guidelines Beachten Sie bitte die unten stehenden Richtlinien		
choisir une réponse select an answer Antwort auswählen	ne PAS choisir une réponse NOT select an answer NICHT Antwort auswählen	Corriger une réponse Correct an answer Antwort korrigieren
  		 
ce qu'il ne faut PAS faire what should NOT be done was man NICHT tun sollte		
     		



Question 14: Cette question est notée sur 2.5 points.

0 .5 1 .5 2 .5

```
1 data = [9, 3, -3, -1, -6]
2
3 for i in range(3):
4     for k in range(len(data) - 1):
5         if data[k] > data[k + 1]:
6             temp = data[k]
7             data[k] = data[k + 1]
8             data[k + 1] = temp
9
10 print(data[1::2])
```

Qu'affiche le programme ci-dessus ?

[-6, 3]

Question 15: Cette question est notée sur 2.5 points.

0 .5 1 .5 2 .5

```
1 import numpy as np
2
3 arr1 = np.arange(-4, 8)
4 arr2 = arr1.reshape(3, 4)
5 arr3 = arr1.reshape(4, 3)
6
7 a = arr2[:, arr3.shape[0] // arr3.shape[1]]
8 b = arr3[arr2.shape[0] // arr2.shape[1], :]
9 print(np.min(a), np.min(b))
```

Qu'affiche le programme ci-dessus ?

-3 -4



Question 18: Cette question est notée sur 2.5 points.

0 .5 1 .5 2 .5

```
1 def foo(x, n):
2     if n == 0:
3         return x * 2 + 1
4     elif n < 0:
5         return foo(x - 1, 0)
6     else:
7         return foo(x + 4, n - 2)
8
9 print(foo(2.5, 5))
```

Qu'affiche le programme ci-dessus ?

28.0

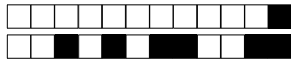
Question 19: Cette question est notée sur 2.5 points.

0 .5 1 .5 2 .5

```
1 l1 = [0, 1, 3, 8, 9, 4]
2 l2 = [1, 2, 6, 9, 1, 3]
3 steps = list(zip(l1, l2))
4
5 my_dict = {}
6 my_new_dict = {}
7
8 for current, next in steps:
9     my_dict[current] = next
10
11 for key in my_dict:
12     val = my_dict[key]
13     while val in my_dict:
14         val = my_dict[val]
15     my_new_dict[key] = val
16
17 print("my_new_dict[4] =", my_new_dict[4])
18 print("my_new_dict[8] =", my_new_dict[8])
```

Qu'affiche le programme ci-dessus ?

my_new_dict[4] = 6
my_new_dict[8] = 2



Question 21: Cette question est notée sur 2.5 points.

0 .5 1 .5 2 .5

```
1 s = "How much wood would a woodchuck chuck if a woodchuck could chuck wood?"
2 t = "woodpecker"
3
4 s_s = list(s)
5 t_s = list(set(t))
6
7 my_dict = {k : s_s.count(k) for k in t_s}
8
9 my_list = []
10 for k, v in my_dict.items():
11     if v == 0:
12         my_list.append(k)
13
14 for k in my_list:
15     my_dict.pop(k)
16
17 print("len(my_list) =", len(my_list))
18 print("len(my_dict) =", len(my_dict))
19 print("my_dict[t[3]] =", my_dict[t[3]])
```

Qu'affiche le programme ci-dessus ?

```
len(my_list) = 3
len(my_dict) = 5
my_dict[t[3]] = 6
```