



1

Ens. : O. Lévêque, P. Wang
Information, Calcul, Communication - A
Vendredi 22 décembre 2023
Durée : 150 minutes













U.N.Owen

SCIPER: 0

Salle: INF 1

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 10 pages, les dernières pouvant être vides. Ne pas dégrafer.

- Posez votre carte d'étudiant sur la table.
- Document autorisé pour cet examen : un formulaire constitué d'une page A4 recto-verso, manuscrite (ou préparée avec stylet+tablette).
- L'utilisation tout appareil électronique (calculatrice, ordinateur, smartphone/watch, tablette) est interdite pendant l'épreuve.
- L'examen est composé de deux parties:
 - une partie avec 14 questions à choix multiple ; chaque question admet une seule réponse correcte parmi 4 possibilités : la réponse correcte vaut 1 point ; toute autre option (pas de réponse, réponse fausse, ou plusieurs cases cochées) vaut 0 point.
 - une partie avec 4 questions de type ouvert, valant en tout 16 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 Observe this 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		
     		



Première partie, questions à choix multiple

Pour chaque question, marquer la case correspondant à la réponse correcte sans faire de ratures. Il n'y a qu'une seule réponse correcte par question.

Question 1

Quelle est la bande passante B du signal X suivant (on suppose ici que le temps t est mesuré en secondes) ?

$$X(t) = \sum_{n=1}^{10} n \cdot \sin\left(\frac{2\pi n t}{n}\right), \quad t \in \mathbb{R}$$

- $B = 10$ Hz $B = 1$ Hz $B = \frac{1}{10}$ Hz $B = +\infty$

Question 2

Pour reconstruire le signal X de la question 1 à partir de sa version échantillonnée, on utilise la formule

$$X_I(t) = \sum_{m \in \mathbb{Z}} X\left(\frac{mT_e}{2}\right) \cdot \text{sinc}\left(\frac{2t - mT_e}{T_e}\right), \quad t \in \mathbb{R}$$

Quelle condition *minimale* la fréquence $f_e = \frac{1}{T_e}$ doit-elle satisfaire pour garantir que $X_I(t) = X(t), \forall t \in \mathbb{R}$?

- Ce n'est jamais le cas que $X_I(t) = X(t), \forall t \in \mathbb{R}$. $f_e > 1$ Hz
 $f_e > 10$ Hz $f_e > 4$ Hz

Question 3

Quelle est l'entropie H de la séquence de 7 lettres ALFALFA ?

- $H = \log_2(7) - \frac{4}{7} \log_2(3) - \frac{4}{7}$ $H = \log_2(7) - \frac{4}{7} \log_2(3) - \frac{8}{7}$
 $H = \log_2(7) - \frac{3}{7} \log_2(3) - \frac{8}{7}$ $H = \log_2(7) - \frac{3}{7} \log_2(3) - \frac{4}{7}$

Question 4

Pour encoder sous forme binaire la séquence de 15 lettres

A H A H A E H E H E I H I H I

quatre dictionnaires sont proposés ci-dessous. Lequel a un défaut ?

- A: 11, E: 10, I: 111, H: 0 A: 11, E: 100, I: 101, H: 0
 A: 11, E: 10, I: 01, H: 00 A: 111, E: 110, I: 10, H: 0

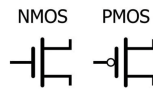
Question 5 En choisissant un des trois bons dictionnaires dans la question 4, calculez le nombre moyen de bits utilisés par lettre pour représenter la séquence AHAEHEHEIHIHI. En vous basant sur ce que vous avez appris en cours (pas besoin d'un calcul explicite), que pouvez-vous déduire sur la valeur de l'entropie H de cette séquence ?

- $2 \leq H \leq 3$ $H \geq 3$ $H \leq 1$ $1 \leq H \leq 2$

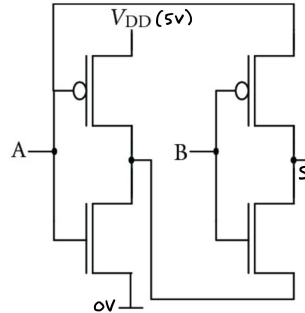


Question 6

Rappel: Un transistor n-mos laisse passer le courant entre l'émetteur et le collecteur lorsqu'une tension haute est présente à la base, tandis qu'un transistor p-mos fait le contraire. Voici aussi leurs symboles respectifs:



Considérons maintenant le circuit suivant:



En fonction des entrées A et B de ce circuit, que vaut la sortie S ?

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

A	B	S
0	0	0
0	1	1
1	0	0
1	1	1

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

A	B	S
0	0	1
0	1	0
1	0	1
1	1	0

Question 7

Considérons un graphe composé de n sommets et au plus n^2 arêtes, toutes de poids 1. Laquelle des affirmations suivantes est-elle correcte ?

- L'algorithme BFS permet de trouver un arbre couvrant minimal du graphe en un temps exponentiel en n .
- L'algorithme BFS permet de trouver un chemin fermé passant une et une seule fois par chaque sommet du graphe en un temps polynomial en n .
- L'algorithme BFS permet de trouver le plus court chemin entre deux sommets A et B du graphe en un temps polynomial en n .
- L'algorithme BFS permet de trouver le plus court chemin entre deux sommets A et B du graphe en un temps exponentiel en n .

Question 8

Un système cryptographique utilise deux clés secrètes K_1 et K_2 , consistant chacune en une séquence de 20 bits. Avec une attaque par force brute sur ce système, un ordinateur de dernière génération prendrait environ 1'000 ans pour trouver les deux clés K_1 et K_2 . Mais si la clé K_2 se trouvait compromise, de combien de temps aurait besoin ce même ordinateur pour trouver la clé K_1 ?

- environ 10 heures
- environ 500 ans
- environ 2,5 ans
- environ 50 ans

Note: On suppose ici qu'une attaque par force brute sur l'une des deux clés sans la connaissance de l'autre clé ne donne rien!

**Question 9**

Examinez le programme ci-après. Combien d'éléments possèdent les ensembles **e** et **f** ?

```
a = list(range(-3, 5))
b = list(range(3, 1))
c = set(a)
d = set(b)
e = c.intersection(d)
f = c.union(d)
```

- e** est vide et **f** possède huit éléments.
- e** possède un élément et **f** possède huit éléments.
- e** possède un élément et **f** possède neuf éléments.
- e** est vide et **f** possède neuf éléments.

Question 10

Quel sera le message affiché suite à l'exécution de ce programme ?

```
d = {k: 2*k+1 for k in range(5)}
output = ""
for i, j in d.items():
    output += str(i) + "," + str(j) + ";"
print(output)
```

- 1,1;3,3;5,5;7,7;9,9;
- 0,2;1,4;2,6;3,8;4,10;
- 0,1;1,3;2,5;3,7;4,9;
- 1,0;3,1;5,2;7,3;9,4;

Question 11

Quel sera le message affiché suite à l'exécution de ce programme ?

```
@dataclass
class ExamClass:
    x: int
    y: int
    z: int

    def do_something(self, x, y):
        self.z = x * y

o = ExamClass(4, 6, 0)
o.do_something(7, 3)
print(o.x, o.y, o.z)
```

- 4 6 21
- 7 3 21
- 7 3 24
- 4 6 24



Question 12 Examinez le programme qui suit :

```
liste = []
for i in range(3):
    l = []
    for j in range(2, 5):
        l.append((i, j))
    liste.append(l)
```

Parmi les extraits suivants, lequel permet de reconstruire exactement le contenu de la variable `liste` construite dans le programme plus haut ?

- `liste = [(x, y) for x in range(3)] for y in range(2, 5)]`
- `liste = [(x, y) for x in range(2, 5)] for y in range(3)]`
- `liste = [(x, y) for y in range(2, 5)] for x in range(3)]`
- `liste = [(x, y) for y in range(2, 5) for x in range(3)]`

Question 13

Quel sera le message affiché à l'exécution de ce programme ?

```
l = list(range(0, 30, 4))
m = list(filter(lambda x: x%3 == 0, l))
print(len(l), len(m))
```

- 8 4
- 8 10
- 8 2
- 8 3

Question 14 Quel sera le message affiché suite à l'exécution de ce programme ?

```
s = "LRL"
G = {"AAA": ("BBB", "CCC"),
     "BBB": ("DDD", "EEE"),
     "CCC": ("ZZZ", "GGG"),
     "DDD": ("DDD", "AAA"),
     "EEE": ("EEE", "BBB"),
     "GGG": ("CCC", "GGG"),
     "ZZZ": ("BBB", "ZZZ")}
```

```
steps = 0
n = "AAA"
while steps < 5:
    if s[steps % len(s)] == 'R':
        n = G[n][1]
    else:
        n = G[n][0]
    steps+=1
print(n)
```

- BBB
- AAA
- DDD
- EEE



Question 18: *Cette question est notée sur 4 points.*

0 1 2 3 4

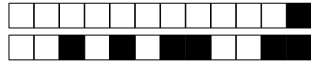
Dans cet exercice, nous travaillons avec des graphes dirigés représentés par des dictionnaires. Dans l'exemple de graphe dirigé ci-après, le nœud *A* possède deux arêtes qui vont vers les nœuds *B* et *C*, *B* possède deux arêtes qui vont vers *C* et *D*, *C* possède deux arêtes qui vont vers *A* et *D*, et *D* possède deux arêtes qui vont vers *A* et *C*:

```
G = {"A": ["B", "C"],  
     "B": ["C", "D"],  
     "C": ["A", "D"],  
     "D": ["A", "C"]}
```

Dans un graphe dirigé, on appelle “degré entrant d’un nœud” le nombre de voisins qui *pointent* vers ce nœud. Dans l'exemple plus haut, les degrés entrants de *A*, *B*, *C*, et *D* sont respectivement 2, 1, 3, et 2.

Dans cet exercice, vous devez implémenter une fonction `in_degree(g: Dict[str, List[str]]) -> Dict[str, int]` qui prend en paramètre un graphe `g` dont la représentation serait semblable à l'exemple donné plus haut, et qui calcule le degré entrant de chaque nœud de `g` puis retourne le tout sous la forme d’un dictionnaire dont les clés sont les noms des nœuds et les valeurs sont les degrés entrants correspondant.

```
def in_degree(g: Dict[str, List[str]]) -> Dict[str, int]:
```



Vous pouvez utiliser cette page au cas où vous n'auriez pas assez de place pour répondre à une question.
Pour faciliter la correction, merci de bien indiquer à la question donnée qu'il faut venir lire la suite ici !

