



1

Ens. : O. Lévêque, J.-Ph. Pellet
Information, Calcul, Communication - A
Vendredi 23 décembre 2022
Durée : 135 minutes

U.N.Owen

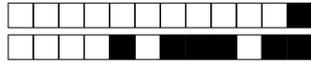
SCIPER: 0

Salle: INF 1

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 7 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 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 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 des questions de type ouvert, valant en tout 14 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 correspondante à la réponse correcte sans faire de ratures. Il n'y a qu'une seule réponse correcte par question.

Question 1 n participants à un concours obtiennent chacun un score allant de 1 à m . Combien de bits sont-ils nécessaires en tout pour enregistrer le nombre de personnes ayant obtenu chaque score ? (on suppose ici que n et m sont des grands nombres)

- $\Theta(m \cdot \log_2(n))$ $\Theta(n \cdot \log_2(m))$ $\Theta(m \cdot n)$ $\Theta(\log_2(m) \cdot \log_2(n))$

Question 2 On considère le signal suivant : $X(t) = \sin(4\pi t) \cdot \sin(6\pi t) \cdot \sin(8\pi t)$, où $t \in \mathbb{R}$ est mesuré ici en secondes. Quelle est la bande passante de X ?

- 8 Hz 12 Hz 4 Hz 9 Hz

Question 3 Soit X un signal dont la bande passante f_{\max} est égale à 25 kHz, et filtré par un filtre passe-bas idéal de fréquence de coupure $f_c = 10$ kHz. On désire enregistrer une seconde de ce signal filtré sur un support numérique en l'échantillonnant, avec chaque échantillon représenté sur 16 bits. Combien de bits au total faut-il enregistrer au minimum si on veut éviter l'effet stroboscopique lors de la reconstruction du signal ?

- 800 kbits 160 kbits 400 kbits 320 kbits

Question 4 Que vaut l'entropie de la séquence de 9 lettres MALAYALAM ?

- $2 \log_2(3) - \frac{4}{3}$ $\log_2(3) + \frac{2}{3}$ $2 \log_2(3) - \frac{8}{9}$ $\log_2(3) + \frac{4}{3}$

Question 5 Si maintenant on encode la séquence MALAYALAM sous la forme d'une séquence binaire, en attribuant à chaque lettre un mot de code selon l'algorithme de Huffman, quel sera le nombre de bits composant la séquence binaire ?

- 19 20 17 18

Question 6 Pour envoyer une direction N, S, E ou W, on utilise le dictionnaire suivant :

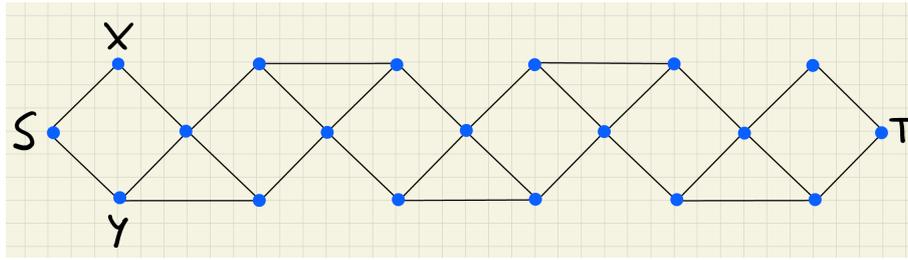
N	S	E	W
1111100	0011111	0111110	0000000

Combien d'effacements ce dictionnaire permet-il de corriger, dans le pire des cas ?

- 2 3 1 4



Question 7 On considère le réseau suivant:



Quelle ligne trouve-t-on dans la table de routage du noeud S ?

Destination	Direction	Distance
T	X	10

Destination	Direction	Distance
T	X	9

Destination	Direction	Distance
T	Y	9

Destination	Direction	Distance
T	Y	10

Question 8 Pour chiffrer un message $X = (X_1, \dots, X_n)$, où chaque $X_i \in \{0, 1, 2, \dots, 9\}$, on utilise le système suivant :

- (a) On tire au hasard une séquence $K = (K_1, \dots, K_n)$, où pour chaque chiffre K_i :
- avec probabilité $1/2$, $K_i = 0$
 - avec probabilité $1/2$, K_i est tiré uniformément au hasard dans l'ensemble $\{1, \dots, 9\}$
- Cette séquence est partagée au préalable avec le destinataire du message.

- (b) Puis on compose le message chiffré $C = (C_1, \dots, C_n)$ suivant : pour chaque valeur de $i \in \{1, \dots, n\}$, $C_i = X_i + K_i \pmod{10}$

Laquelle des affirmations ci-dessous est-elle vraie?

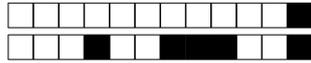
- Ce système de chiffrement est parfaitement sûr, car pour chaque $i \in \{1, \dots, n\}$ la probabilité de changer la valeur de X_i vaut $1/2$.
- Ce système de chiffrement est parfaitement sûr, car la probabilité que le message chiffré C soit égal au message d'origine X est très faible si n est grand.
- Ce système de chiffrement n'est pas parfaitement sûr, car la probabilité que le message chiffré C soit égal au message d'origine X n'est pas nulle.
- Pour obtenir un système de chiffrement parfaitement sûr, il faudrait plutôt poser, pour chaque $i \in \{1, \dots, n\}$: $C_i = X_i + L_i \pmod{10}$, avec L_i tiré uniformément au hasard dans l'ensemble $\{0, 1, 2, \dots, 9\}$.

Question 9

On considère l'extrait de code ci-dessous. Qu'affiche-t-il lors de son exécution?

```
values = [[i * j for i in range(j, 1)] for j in range(5)]
print(values)
```

- `[[0, 0, 0, 0, 0], [1, 2, 3, 4], [4, 6, 8], [9, 12], [16]]`
- `[[0], [1], [4], [9], [16]]`
- `[[0], [1], [1], [1], [1]]`
- `[[1], [0], [0, 2], [0, 3, 6], [0, 4, 8, 12]]`

**Question 10**

Laquelle de ces boucles n'affichera **pas** les index des éléments d'une `List[int]` stockée par la variable `numbers`?

A)

```
for i in range(len(numbers)):
    print(i)
```

B)

```
for i, elem in enumerate(numbers):
    print(i)
```

C)

```
for i in numbers:
    print(i)
```

D)

```
i: int = 0
while i < len(numbers):
    print(i)
    i += 1
```

 Extrait A) Extrait B) Extrait C) Extrait D)**Question 11**

On considère l'extrait de code ci-dessous. Qu'affiche-t-il lors de son exécution?

```
def run_twice(f: Callable[[int], int], x: int) -> int:
    print(f(f(x)))
```

```
def square(x: int) -> int:
    return x * x
```

```
run_twice(square(2), 4)
```

 Une erreur 4 256 16

4 (sur deux lignes)

Question 12

On considère le code lacunaire ci-dessous:

```
___ condition1():
    print("A")
```

```
___ condition1() and condition2():
    print("B")
```

Si `condition1()` et `condition2()` sont les deux évaluées à `True`, quels doivent être, dans l'ordre, les mots clés manquants pour afficher à la fois **A** et **B** lors de l'exécution?

 `if — else` `if — elif` `if — if` `if — else if`**Question 13**

Quelle affirmation sur les threads et les locks est incorrecte?

 Lorsqu'un thread cherche à déverrouiller un verrou, cela peut le bloquer. Un thread peut lui-même créer un autre thread. Lorsqu'un thread cherche à verrouiller un verrou, cela peut le bloquer. Les threads permettent d'exécuter plusieurs séquences d'instructions en parallèle (ou avec un système *round-robin*)



Question 14

Que fait cet extrait de code avec les lignes du fichier `data.txt`? Dans les réponses ci-dessous, on considère comme *mot d'une ligne* une suite de caractères séparées d'une autre.

```
with open("data.txt", "r", encoding="utf-8") as file:
    for line in file.read().split("\n"):
        parts = line.split(" ")
        if len(parts) > 2:
            parts = parts[::-1]
        print(" ".join(parts))
```

- Pour chaque ligne, afficher les mots dans l'ordre inverse s'il y a plus de deux mots et dans l'ordre du fichier sinon.
- Pour chaque ligne, afficher les deux premiers mots dans l'ordre inverse.
- Pour chaque ligne qui a plus de deux mots, afficher les mots dans l'ordre inverse.
- Pour chaque ligne, afficher le dernier mot s'il y a plus de deux mots sur la ligne.

Deuxième partie, questions de type ouvert

Question 15: *Cette question est notée sur 2 points.*

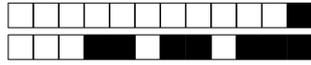
0 1 2

Implémentez la fonction `get_chunk` de manière à retourner le `i`^e groupe de `size` caractères du paramètre `msg`. Le premier groupe a l'index 0. Si `i` est hors limite pour retourner une chaîne de `size` caractères, retournez une chaîne vide (`""`). Exemples:

```
get_chunk("abcdef", i=0, size=2)      doit retourner "ab"
get_chunk("1234567890", i=2, size=3)  doit retourner "789"
get_chunk("1001011001101001", i=2, size=4) doit retourner "0110"
get_chunk("hello", i=1, size=3)       doit retourner ""
```

```
def get_chunk(msg: str, i: int, size: int) -> str:
```

Réponse: Voir <https://moodle.epfl.ch/mod/page/view.php?id=1200738>.

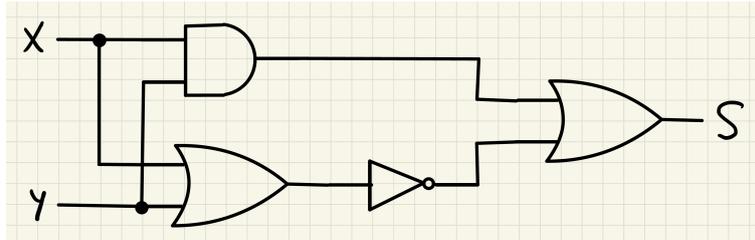


Question 16: Cette question est notée sur 4 points.

0 1 2 3 4

Construire un circuit logique avec des portes ET, OU et NON (et uniquement celles-ci), qui prend en entrée deux bits X et Y , et dont le bit de sortie S vaut 1 si et seulement si $X = Y = 1$ ou $X = Y = 0$. Le maximum de 4 points est obtenu si votre circuit comporte au plus 4 portes (et que sa sortie est correcte). Chaque porte additionnelle entraîne une déduction de 1 point.

Réponse: Voici un circuit possible:



Question 17: Cette question est notée sur 4 points.

0 1 2 3 4

Rappel : Pour $a, b \in \mathbb{R}$, $\sin(a) - \sin(b) = 2 \cdot \sin\left(\frac{a-b}{2}\right) \cdot \cos\left(\frac{a+b}{2}\right)$.

On considère le signal $X(t) = \sin(2\pi ft)$, où f est une fréquence donnée (mesurée en Hz), ainsi que le signal $\hat{X}(t) = X(t) - X(t - t_0)$, où $t_0 > 0$ est un intervalle de temps donné (mesuré en secondes).

a) Quelle est la bande passante du signal \hat{X} ?

Réponse: La bande passante vaut f , car chaque composante est de fréquence f (et les deux composantes ne s'annulent pas).

b) Exprimer l'amplitude du signal \hat{X} en fonction de f et t_0 .

Réponse: En utilisant la formule donnée en indication, on trouve

$$\hat{X}(t) = 2 \cdot \sin(\pi f t_0) \cdot \cos(2\pi f t - \pi f t_0)$$

et donc, l'amplitude de ce signal vaut $a = 2 \cdot |\sin(\pi f t_0)|$ (réponse $a = 2 \cdot \sin(\pi f t_0)$ aussi acceptée).

c) Donner une valeur de t_0 (dépendant de f) pour laquelle l'amplitude du signal \hat{X} est maximale.

Réponse: L'amplitude est maximale quand par exemple $\pi f t_0 = \pi/2$, i.e. $t_0 = 1/2f$.

d) La transformation du signal X en un signal \hat{X} peut-elle être interprétée comme un filtre passe-bas? Justifier votre réponse.

Réponse: Non: l'amplitude de sortie oscille avec la valeur de f en entrée, mais ne décroît pas lorsque f grandit (à noter qu'il est faux de dire que ce n'est pas un filtre passe-bas car la fréquence de sortie reste identique; en effet, le filtre à moyenne mobile, qui lui est un filtre passe-bas, a cette même propriété).



Question 18: Cette question est notée sur 4 points.

0
 1
 2
 3
 4

On considère la séquence de 36 bits suivante :

000001010100 000010001100 000100010001

(les espaces ci-dessus sont juste ajoutés pour la lisibilité de la séquence, mais sont à ignorer)

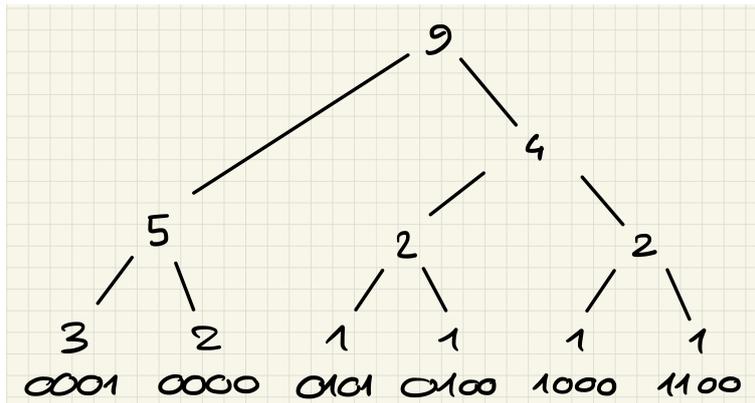
Pour compresser cette séquence, on considère deux stratégies possibles :

- (a) Regrouper les bits par groupes de 4, en considérant chaque groupe de 4 bits comme un seul symbole, et appliquer l'encodage de Huffman à ces symboles.
- (b) Regrouper les bits par groupes de 3, en considérant chaque groupe de 3 bits comme un seul symbole, et appliquer l'encodage de Huffman à ces symboles.

Etablir le code de Huffman dans chacun des deux cas, et déterminer quel code mène à une meilleure compression de la séquence d'origine.

Réponse:

- En regroupant les bits par 4, on obtient l'arbre suivant avec l'algorithme de Huffman:



ce qui donne le dictionnaire:

symbole	0001	0000	0101	0100	1000	1100
nb apparitions	3	2	1	1	1	1
mot de code	11	10	011	010	001	000

Et donc le nombre total de bits utilisés dans ce cas vaut $5 \cdot 2 + 4 \cdot 3 = 22$.

- En regroupant les bits par 3, les symboles 000, 001, 010 et 100 apparaissent chacun 3 fois dans la séquence, et le codage de Huffman utilise 2 bits pour chaque symbole, donc $4 \cdot 3 \cdot 2 = 24$ bits au total. La première méthode est donc plus efficace.