Corrected

EPFL

Enseignant : Rafael Pires
CS-119(k) ICC Final - CS
26.06.2026 9h15
180 min.

1

Tartempion Machinchose

SCIPER : **000000** Salle : **PO 01** Signature :

Attendez le début de l'épreuve avant de tourner la page. Ce document est imprimé recto-verso, il contient 20 pages, les dernières pouvant être vides. Ne pas dégrafer.

- Posez votre carte d'étudiant sur la table.
- Deux pages A4 (recto-verso) de résumé personnel sont autorisées. Aucun dispositif de magnification.
- L'utilisation d'une **calculatrice** et de tout outil électronique est interdite pendant l'épreuve.
- Première partie **(60 points)** : questions à choix multiple, avec **une ou plusieurs réponses correctes** à chaque question (4 points par question) :
  - **Pour chaque case**, on comptera +1 point si son état (cochée ou non) est correct et −1 point sinon,

  ≥ 0 minimum 0 point à chaque question,

  0 point à la question si aucune case n'est cochée.
- Deuxième partie **(20 points)** : Pour les questions **vrai/faux**, on comptera,:

  +1 points si la réponse est correcte,

  0 point si il n'y a aucune ou plus d'une réponse inscrite,

  −1 point si la réponse est incorrecte.

  ≥ 0 le total de la deuxième partie ne peut pas être négatif.
- Troisième partie **(30 points)** : pour les questions ouvertes, le nombre de point maximum est noté au-dessus de chaque question. Laissez les cases à cocher vides !
- Utilisez un **stylo** à encre **noire ou bleu foncé** et effacez proprement avec du **correcteur blanc** si nécessaire.
- Si une question est erronée, l'enseignant se réserve 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

Seules les réponses inscrites dans le cahier agrafé en français seront considérées ; les réponses sur le supplément de traduction anglaise, les brouillons ou feuilles volantes ne seront pas pris en compte.

CORRECTED

Première partie, questions à choix multiple

Pour chaque question marquer la/les case(s) correspondante(s) à la/aux réponse(s) correcte(s) sans faire de ratures.

Question 1
[4 points] On considère le modèle TCP/IP en 5 couches vu en cours. Cochez **toutes** les affirmations correctes :

- [x] Le routage des paquets de bout en bout est le rôle de la couche Réseau (protocole IP).
- [x] Les protocoles TCP et UDP appartiennent à la couche Transport.
- [ ] Le protocole HTTP appartient à la couche Liaison.
- [ ] L'adresse MAC sert au routage de bout en bout à travers Internet.

Question 2
[4 points] On exécute le code Python suivant :

```python
i = 0
s = 0
while True:
    i += 1
    if i % 2 == 0:
        continue
    if i > 7:
        break
    s += i
print(s)
```

Cochez **toutes** les affirmations correctes :

- [x] À la sortie de la boucle, `i` vaut `9`.
- [x] Le programme affiche `16`.
- [ ] La boucle ne termine jamais, car la condition du `while` est `True`.
- [ ] L'instruction `continue` termine définitivement la boucle.

Question 3
[4 points] Cochez **toutes** les affirmations correctes concernant les fonctions de hachage et la cryptographie asymétrique :

- [x] Une signature numérique se crée avec la clé privée du signataire et se vérifie avec sa clé publique.
- [ ] Pour envoyer un message confidentiel à Bob, Alice le chiffre avec sa propre clé publique.
- [ ] Une fonction de hachage cryptographique permet de retrouver facilement le message original à partir de son empreinte.
- [x] Pour stocker des mots de passe, il vaut mieux enregistrer `hash(sel + mot_de_passe)` que le mot de passe en clair.

Question 4
[4 points] On chiffre le message $M = \texttt{10110010}$ avec la clé $K = \texttt{01101011}$ à l'aide d'un masque jetable (*one-time pad*) : $C = M \oplus K$ (XOR bit à bit). Cochez **toutes** les affirmations correctes :

- [x] Le destinataire, qui connaît K, retrouve le message en calculant $C \oplus K$.
- [ ] Le texte chiffré est $C = \texttt{11111011}$.
- [ ] La sécurité reste parfaite même si la même clé K est réutilisée pour chiffrer un second message.
- [x] Le texte chiffré est $C = \texttt{11011001}$.

CORRECTED

Question 5

[4 points] Cochez **toutes** les affirmations correctes concernant les *threads* en Python (module `threading`) :

- ☐ C'est la méthode `join()` qui démarre l'exécution d'un thread.
- ■ L'ordre exact d'exécution de plusieurs threads n'est pas prévisible : il est géré par le système d'exploitation.
- ☐ Un programme multi-thread sans verrou produit toujours un résultat incorrect.
- ■ Sans verrou (`Lock`), deux threads qui modifient une même variable partagée peuvent provoquer une condition de course (*race condition*).

Question 6

[4 points] On exécute le code Python suivant :

```
mots = ["pomme", "kiwi", "abricot", "figue"]
mots.sort(key=len)
```

Cochez **toutes** les affirmations correctes :

- ■ La méthode `sort` modifie la liste `mots` en place.
- ■ `mots[-1]` vaut `"abricot"`.
- ☐ `sort(key=len)` trie les mots par ordre alphabétique.
- ■ `mots[0]` vaut `"kiwi"`.

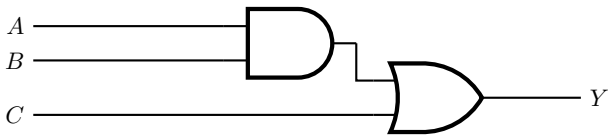
Question 7

[4 points] Cochez **toutes** les affirmations correctes concernant la communication sur Internet :

- ☐ Internet repose sur la commutation de circuits : une ligne dédiée est réservée pour chaque communication.
- ☐ Le protocole UDP garantit que les paquets arrivent à destination et dans l'ordre.
- ■ Un message est découpé en paquets qui peuvent suivre des chemins différents et arriver dans le désordre.
- ■ Le protocole TCP assure une livraison fiable grâce aux acquittements (ACK) et aux retransmissions.

Question 8

[4 points] Pour le circuit logique ci-dessous, cochez **toutes** les affirmations correctes :



- ■ Si $C = 1$, alors $Y = 1$ quelles que soient les valeurs de A et B.
- ☐ La sortie vaut $Y = 1$ si et seulement si $A = B = C = 1$.
- ■ Si $A = 0$ et $C = 0$, alors $Y = 0$ quelle que soit la valeur de B.
- ☐ Le circuit calcule $Y = A$ ET B ET C.

Question 9
[4 points] On exécute le code Python suivant :

```python
with open("notes.txt", "w") as f:
    f.write("12\n")
    f.write("15\n")
    f.write("9\n")

total = 0
with open("notes.txt", "r") as f:
    for ligne in f:
        total += int(ligne.strip())
print(total)
```

Cochez **toutes** les affirmations correctes :

- [x] Le fichier est automatiquement fermé à la fin de chaque bloc `with`.
- [ ] La boucle `for ligne in f` parcourt le fichier caractère par caractère.
- [x] Le programme affiche `36`.
- [x] Après le premier bloc `with`, le fichier `notes.txt` contient trois lignes.

Question 10
[4 points] Une source émet l'un de 16 symboles, tous équiprobables. Cochez **toutes** les affirmations correctes :

- [x] L'entropie de la source vaut 4 bits.
- [ ] Si la distribution des symboles n'était pas uniforme, l'entropie serait plus grande.
- [x] Il faut en moyenne, au minimum, 4 questions binaires (oui/non) pour identifier le symbole émis.
- [x] Un code de longueur fixe à 4 bits par symbole est ici optimal.

Question 11
[4 points] On échantillonne un signal $X(t)$ de bande passante $f_{\max}$ à la fréquence f_e. Cochez **toutes** les affirmations correctes :

- [x] Si $f_e > 2f_{\max}$, l'interpolation sinc reconstruit parfaitement le signal.
- [x] Appliquer un filtre passe-bas idéal de fréquence de coupure $f_e/2$ avant d'échantillonner permet d'éviter le repliement spectral.
- [ ] Augmenter f_e au-delà de $2f_{\max}$ dégrade la qualité de la reconstruction.
- [x] Si $f_e < 2f_{\max}$, un effet stroboscopique (repliement spectral) peut apparaître.

Question 12

[4 points] La variable `img` contient une image en niveaux de gris, représentée comme dans le miniprojet par une liste de listes d'entiers entre 0 et 255 (au moins un pixel). On définit :

```python
def mystere(img: list[list[int]]) -> int:
    total = 0
    n = 0
    for ligne in img:
        for pixel in ligne:
            total += pixel
            n += 1
    return total // n
```

Cochez **toutes** les affirmations correctes :

- [x] `mystere(img)` retourne la valeur moyenne (arrondie vers le bas) des pixels de l'image.
- [ ] Le résultat retourné est de type `float`.
- [x] Le résultat est toujours compris entre 0 et 255.
- [ ] `mystere(img)` retourne la somme de tous les pixels de l'image.

Question 13

[4 points] Soit le signal $X(t) = \sin(6\pi t) + \cos(10\pi t)$. Cochez **toutes** les affirmations correctes :

- [x] La bande passante de $X(t)$ vaut 5 Hz.
- [x] Échantillonner à $f_e = 11$ Hz garantit une reconstruction parfaite par interpolation sinc.
- [ ] Échantillonner à $f_e = 10$ Hz garantit une reconstruction parfaite par interpolation sinc.
- [x] Les fréquences des composantes de $X(t)$ sont 3 Hz et 5 Hz.

Question 14

[4 points] On exécute le code Python suivant :

```python
L = [x * x for x in range(6) if x % 2 == 1]
```

Cochez **toutes** les affirmations correctes :

- [ ] `L` contient les carrés de tous les entiers de 0 à 5.
- [ ] Ce code lève une erreur : une compréhension de liste ne peut pas contenir de condition `if`.
- [x] `L` vaut `[1, 9, 25]`.
- [x] `len(L)` vaut `3`.

Question 15

[4 points] On considère le circuit logique qui calcule $S = (\text{NON } A) \text{ OU } (\text{NON } B)$. Cochez **toutes** les affirmations correctes :

- [x] $S = 0$ si et seulement si $A = B = 1$.
- [ ] S est équivalent à NON(A OU B), c'est-à-dire à une porte NOR.
- [ ] $S = 1$ si et seulement si A et B sont différents.
- [x] S est équivalent à NON(A ET B), c'est-à-dire à une porte NAND.

CORRECTED

Deuxième partie, questions du type Vrai ou Faux

Pour chaque question, marquer (sans faire de ratures) la case VRAI si l'affirmation est **toujours vraie** ou la case FAUX si elle **n'est pas toujours vraie** (c'est-à-dire si elle est parfois fausse).

Question 16 Sur Internet, tous les paquets d'un même message suivent nécessairement le même chemin entre l'expéditeur et le destinataire.

☐ VRAI ■ FAUX

Question 17 La sécurité du protocole d'échange de clés de Diffie-Hellman repose sur la difficulté du problème du logarithme discret.

■ VRAI ☐ FAUX

Question 18 Dans le miniprojet, le flou de boîte d'une image est obtenu en appliquant deux convolutions à une dimension successives : une sur les lignes, puis une sur les colonnes.

■ VRAI ☐ FAUX

Question 19 Les formats JPEG (images) et MP3 (audio) sont des exemples de compression sans pertes.

☐ VRAI ■ FAUX

Question 20 Le code `d = {}` suivi de `d["x"] += 1` provoque une erreur à l'exécution.

■ VRAI ☐ FAUX

Question 21 Si l'on ajoute un sel (*salt*) aléatoire propre à chaque utilisateur avant le hachage, deux utilisateurs ayant le même mot de passe obtiennent des hachés différents.

■ VRAI ☐ FAUX

Question 22 Appeler la méthode `start()` d'un `Thread` bloque le programme principal jusqu'à ce que ce thread ait terminé son exécution.

☐ VRAI ■ FAUX

Question 23 La bande passante de la sinusoïde $a \cdot \sin(2\pi f t + \delta)$ dépend de son amplitude a.

☐ VRAI ■ FAUX

Question 24 La sortie d'une porte ET à deux entrées vaut 1 dès qu'au moins une de ses entrées vaut 1.

☐ VRAI ■ FAUX

Question 25 La construction `with open(...) as f:` garantit la fermeture automatique du fichier à la fin du bloc.

■ VRAI ☐ FAUX

Question 26 Dans la cryptographie asymétrique, on déchiffre les messages que l'on reçoit avec sa clé publique.

☐ VRAI ■ FAUX

Question 27 L'ensemble `{x % 3 for x in range(10)}` contient exactement trois éléments.

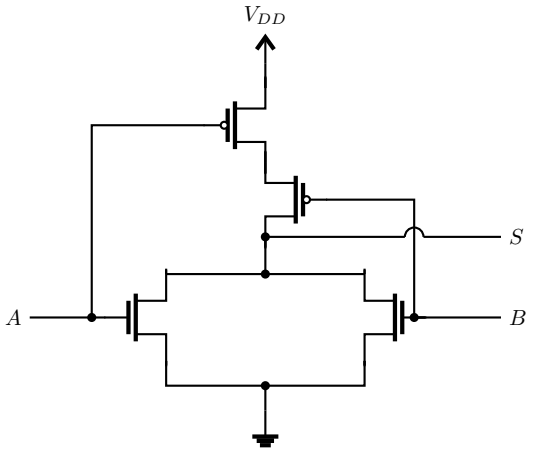
■ VRAI ☐ FAUX

CORRECTED

Question 28 L'entropie d'une séquence composée d'une seule lettre répétée est nulle.

■ VRAI ☐ FAUX

Question 29 Pour le circuit à transistors CMOS ci-dessous, la sortie S vaut 1 uniquement lorsque $A = B = 1$.



☐ VRAI ■ FAUX

Question 30 La compréhension de liste `[2 * x for x in L]` modifie la liste `L` en place.

☐ VRAI ■ FAUX

Question 31 Dans un code préfixe (comme celui produit par l'algorithme de Huffman), aucun mot de code n'est le préfixe d'un autre mot de code.

■ VRAI ☐ FAUX

Question 32 Dans un additionneur 8 bits pour entiers non signés, une retenue finale égale à 1 signale un dépassement de capacité (*overflow*).

■ VRAI ☐ FAUX

Question 33 Échantillonner un signal de bande passante $f_{\max}$ exactement à $f_e = 2f_{\max}$ garantit toujours une reconstruction parfaite par interpolation sinc.

☐ VRAI ■ FAUX

Question 34 La recherche dichotomique ne fonctionne correctement que sur une liste triée.

■ VRAI ☐ FAUX

Question 35 Sans mécanisme de verrou, le résultat final de deux threads qui incrémentent un même compteur partagé peut varier d'une exécution à l'autre.

■ VRAI ☐ FAUX

Troisième partie, questions de type ouvert

Répondre dans l'espace dédié. Votre réponse doit être soigneusement justifiée, toutes les étapes de votre raisonnement doivent figurer dans votre réponse. Laisser libres les cases à cocher : elles sont réservées au correcteur.

Question 36: *Cette question est notée sur 6 points.*

On veut **simuler** le remplissage d'un album d'autocollants de la Coupe du Monde. L'album complet comporte n autocollants différents, numérotés de 0 à $n-1$. Les autocollants se vendent en **sachets** : chaque sachet contient 7 autocollants **aléatoires** (chaque numéro étant équiprobable, et un même autocollant pouvant apparaître plusieurs fois) et coûte 2 CHF. On achète des sachets un à un jusqu'à avoir complété l'album, et on veut connaître le coût total (en CHF) nécessaire.

Indice : `random.randint(a, b)` retourne un entier aléatoire compris entre `a` et `b` **inclus**. On suppose que le module `random` a déjà été importé.

a) (2 points) Écrivez une fonction `acheter_sachet(n)` qui retourne la **liste** des 7 numéros (chacun entre 0 et $n-1$) des autocollants contenus dans un sachet tiré au hasard.

```
1 def acheter_sachet(n: int) -> list[int]:
```

Corrigé.

```
1 def acheter_sachet(n: int) -> list[int]:
2     sachet = []
3     for _ in range(7):
4         sachet.append(random.randint(0, n - 1))
5     return sachet
```

b) (3 points) Écrivez une fonction `cout_pour_completer(n)` qui simule l'achat de sachets (en appelant `acheter_sachet` de la partie a)) jusqu'à posséder les n autocollants différents, et qui retourne le **coût total en CHF** dépensé. Rappel : un sachet coûte 2 CHF. Utilisez un `set` pour mémoriser les autocollants déjà possédés.

```python
def cout_pour_completer(n: int) -> int:
```

Corrigé.

```python
def cout_pour_completer(n: int) -> int:
    possedes = set()
    cout = 0
    while len(possedes) < n:
        sachet = acheter_sachet(n)
        cout += 2
        for autocollant in sachet:
            possedes.add(autocollant)
    return cout
```

Accepter toute variante correcte. Pièges : oublier le +2 CHF par sachet, ou tester `len(possedes) == n` avant le premier achat.

c) (1 point) Dans la partie b), on teste de façon répétée si un autocollant est déjà possédé. Expliquez **brièvement** pourquoi l'utilisation d'un `set` pour stocker les autocollants possédés est plus efficace que celle d'une `list` pour ce test d'appartenance.

Corrigé. Le test d'appartenance `v in possedes` est en $\Theta(1)$ (en moyenne) sur un `set` (table de hachage), contre $\Theta(k)$ sur une `list` de k éléments (parcours linéaire). Cf. transparent « Sets ».

Pour faciliter la correction, **merci de bien indiquer** *à la question donnée qu'il faut venir lire la suite ici !*

CORRECTED

Question 37: *Cette question est notée sur 8 points.*

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ■ 8

On considère le signal suivant :

$$X(t) = 3\sin(8\pi t) + \sin\Big(2\pi t + \frac{\pi}{3}\Big) + 2\cos(5\pi t)$$

a) (1 point) Donnez les fréquences (en Hz) des différentes composantes sinusoïdales de $X(t)$, ainsi que la bande passante $f_{\max}$ du signal.

Corrigé. On identifie $2\pi f$ au coefficient de t : $f_1 = \mathbf{4}$ Hz (de $8\pi t$), $f_2 = \mathbf{1}$ Hz (de $2\pi t$), $f_3 = \mathbf{2{,}5}$ Hz (de $5\pi t$). Bande passante : $f_{\max} = \mathbf{4}$ Hz.

b) (2 points) On fait passer $X(t)$ par un **filtre passe-bas idéal** de fréquence de coupure $f_c = 3$ Hz. Donnez l'expression du signal $Y(t)$ en sortie du filtre et sa bande passante.

Corrigé. Le passe-bas idéal $f_c = 3$ Hz ne laisse passer que les composantes $f < 3$ Hz :

$$Y(t) = \sin\big(2\pi t + \tfrac{\pi}{3}\big) + 2\cos(5\pi t),$$

bande passante = **2,5** Hz (la composante à 4 Hz est supprimée).

c) (1 point) On échantillonne maintenant le signal $X(t)$ **original** (sans filtre). Quelle condition la fréquence d'échantillonnage f_e doit-elle satisfaire pour que l'interpolation sinc reconstruise parfaitement $X(t)$? Justifiez en citant le théorème utilisé.

Corrigé. Théorème d'échantillonnage de Nyquist–Shannon : il faut $f_e > 2f_{\max} = \mathbf{8}$ Hz (inégalité **stricte**) pour que l'interpolation sinc reconstruise $X(t)$ parfaitement.

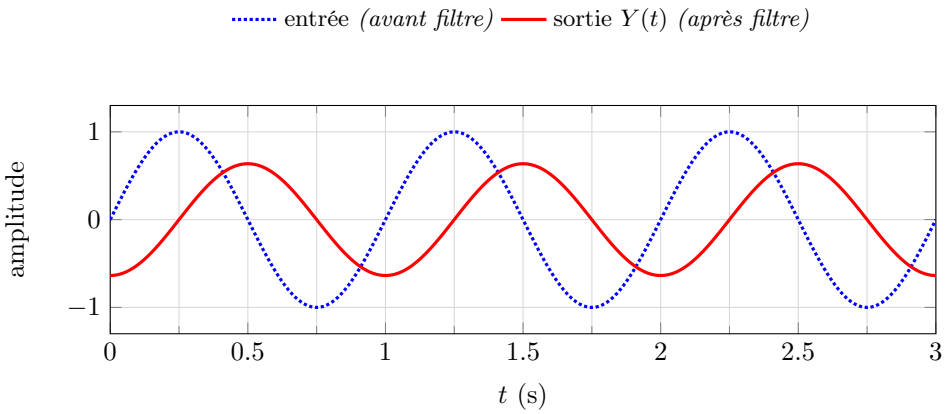
d) (2 points) On échantillonne $X(t)$ à $f_e = 6$ Hz, sans filtrage préalable. Expliquez ce qui se passe lors de la reconstruction (nom du phénomène et composante(s) concernée(s)), puis indiquez comment éviter ce phénomène si l'on doit garder $f_e = 6$ Hz.

Corrigé. À $f_e = 6$ Hz, on a $f_e/2 = 3 < 4$ Hz : la composante à **4 Hz** viole Nyquist $\Rightarrow$ **repliement spectral (effet stroboscopique)** : elle est reconstruite à $f_e - f = 6 - 4 = \mathbf{2}$ Hz. (Les composantes à 1 et 2,5 Hz sont correctes.) Remède en gardant $f_e = 6$ Hz : appliquer un **filtre passe-bas anti-repliement** de coupure ≤ 3 Hz **avant** l'échantillonnage (on perd la composante à 4 Hz, mais plus aucune fréquence fantôme).

e) (2 points) On filtre $X(t)$ par un **filtre à moyenne mobile** de largeur $T_c = 0{,}5$ s, défini par

$$Y(t) = \frac{1}{T_c} \int_{t-T_c}^{t} X(s)\,\mathrm{d}s.$$

Le graphe ci-dessous montre une sinusoïde en entrée (pointillé) et la sortie correspondante $Y(t)$ (trait plein). **(i)** Déterminez, à partir du graphe, le **décalage temporel** (retard) introduit par le filtre. **(ii)** Citez un **autre effet** du filtre sur le signal (visible sur le graphe) et **justifiez-le** (une justification formelle *ou* intuitive sera acceptée).



Corrigé. **(i)** La sortie est décalée d'un quart de période vers la droite : **retard** $\approx 0{,}25$ **s** ($= T_c/2$; la justification du retard n'est pas demandée). **(ii)** Autre effet : l'**amplitude est réduite** (de 1 à $\approx 0{,}64$) — le filtre **lisse** / **atténue** le signal. Justification acceptée : *intuitive* — la moyenne mobile combine sur la fenêtre des valeurs hautes et basses, ce qui « arrondit » les sommets ; *ou formelle* — l'amplitude est multipliée par $|\mathrm{sinc}(fT_c)| = \left|\frac{\sin(\pi f T_c)}{\pi f T_c}\right| < 1$ (ici $\approx 0{,}64$).

CORRECTED

Question 38: *Cette question est notée sur 8 points.*

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ■ 8

On considère la séquence de 16 lettres suivante :

`AABACABDABACABDA`

Rappel : $\log_2 2 = 1$, $\log_2 4 = 2$, $\log_2 8 = 3$.

a) (2 points) Donnez la probabilité d'apparition de chaque lettre dans la séquence, puis calculez l'entropie H de la séquence (en bits).

Corrigé. Comptes : $A \times 8$, $B \times 4$, $C \times 2$, $D \times 2$ (sur 16). Donc $p(A) = \frac{1}{2}$, $p(B) = \frac{1}{4}$, $p(C) = p(D) = \frac{1}{8}$.

$$H = \tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 2 + \tfrac{1}{8}\cdot 3 + \tfrac{1}{8}\cdot 3 = 0{,}5 + 0{,}5 + 0{,}375 + 0{,}375 = \mathbf{1{,}75} \text{ bits.}$$

b) (3 points) Construisez un code de Huffman pour cette séquence. Donnez l'arbre (ou les étapes de construction) et le mot de code binaire associé à chaque lettre.

Corrected

Corrigé. Huffman : fusion $C(2)+D(2)\to CD(4)$; $CD(4)+B(4)\to BCD(8)$; $BCD(8)+A(8)\to$ racine (16). Code (à inversion 0/1 près) :

$$\mathbf{A=0},\quad \mathbf{B=10},\quad \mathbf{C=110},\quad \mathbf{D=111}.$$

c) (2 points) Calculez la longueur moyenne L (en bits par lettre) de votre code de Huffman, puis le nombre total de bits nécessaires pour encoder la séquence. Comparez avec un code de longueur fixe pour ces 4 lettres.

Corrigé. $L = \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 = \mathbf{1{,}75}$ bits/lettre. Total : $16 \times 1{,}75 = \mathbf{28}$ bits, contre $16 \times 2 = \mathbf{32}$ bits pour un code fixe à 2 bits — gain de 4 bits (12,5%).

d) (1 point) Que dit le théorème de Shannon sur la longueur moyenne L de tout code sans pertes par rapport à l'entropie H ? Votre code de Huffman est-il optimal ici ?

Corrigé. Théorème de Shannon : tout code sans pertes vérifie $\mathbf{L \geq H}$. Ici $L = H = 1{,}75 \Rightarrow$ le code de Huffman est **optimal** (probabilités puissances de $\frac{1}{2}$).

Corrected

Pour faciliter la correction, **merci de bien indiquer** *à la question donnée qu'il faut venir lire la suite ici !*

CORRECTED

Question 39: *Cette question est notée sur 8 points.*

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ■ 8

Alice et Bob veulent établir un secret partagé à l'aide du protocole de **Diffie-Hellman**. Ils se mettent d'accord **publiquement** sur le nombre premier $P = 23$ et sur le nombre $Q = 5$.

Note : quelques valeurs de $a^b \bmod 23$:

$5^3 \equiv 10 \pmod{23}$ $\quad 5^4 \equiv 4 \pmod{23}$ $\quad 5^7 \equiv 17 \pmod{23}$ $\quad 5^{12} \equiv 18 \pmod{23}$

$4^3 \equiv 18 \pmod{23}$ $\quad 4^4 \equiv 3 \pmod{23}$ $\quad 10^3 \equiv 11 \pmod{23}$ $\quad 10^4 \equiv 18 \pmod{23}$

a) (2 points) Décrivez les étapes du protocole de Diffie-Hellman : ce que chacun choisit, calcule et envoie, et comment chacun obtient le secret partagé K.

Corrigé. Alice choisit un secret N_A, calcule $Q^{N_A} \bmod P$ et l'envoie ; Bob choisit N_B, calcule $Q^{N_B} \bmod P$ et l'envoie. Alice calcule $(Q^{N_B})^{N_A} \bmod P$, Bob calcule $(Q^{N_A})^{N_B} \bmod P$; les deux valent $Q^{N_A N_B} \bmod P = K$.

b) (3 points) Alice choisit le nombre secret $N_A = 4$ et Bob choisit le nombre secret $N_B = 3$. Donnez les valeurs **échangées** par Alice et Bob, puis calculez le secret partagé K (les deux calculs doivent donner la même valeur).

Corrected

Corrigé. Alice envoie $5^4 \bmod 23 = \mathbf{4}$; Bob envoie $5^3 \bmod 23 = \mathbf{10}$. Alice : $10^4 \bmod 23 = \mathbf{18}$; Bob : $4^3 \bmod 23 = \mathbf{18}$. Secret partagé : $\mathbf{K = 18}$ ($= 5^{12} \bmod 23$). ($5^7 = 17$ est un distracteur : $4 + 3 \neq 4 \times 3$.)

c) (2 points) Eve espionne le canal et intercepte P, Q ainsi que les deux valeurs échangées. Expliquez pourquoi elle ne peut pas (en pratique) calculer le secret K, et pourquoi il faut tout de même utiliser de très grands nombres en pratique.

Corrigé. Eve connaît P, Q, 4 et 10, mais retrouver N_A à partir de $Q^{N_A} \bmod P$ est le problème du **logarithme discret**, difficile (pas d'algorithme efficace connu). Avec de petits nombres, une recherche exhaustive suffit — d'où l'usage de **très grands nombres** (centaines de chiffres) en pratique.

d) (1 point) Une fois le secret K établi, quelle technique cryptographique Alice et Bob peuvent-ils utiliser avec K comme clé partagée ? Nommez-la et citez quelques exemples.

Corrigé. Le **chiffrement symétrique** : K sert de clé partagée pour chiffrer la suite de la communication. Exemples acceptés (vus en cours) : **chiffre de César**, **masque jetable** (one-time pad), **AES**. Le seul « pour échanger des messages » (but général, sans nommer la technique) ne vaut pas le point.

Corrected

Pour faciliter la correction, **merci de bien indiquer** *à la question donnée qu'il faut venir lire la suite ici !*