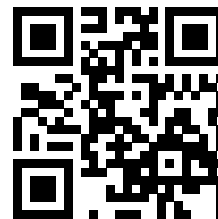


NOM : Hanon Ymous
(000000)
Place : 0

#0000



Information, Calcul et Communication (SPH) : Examen final

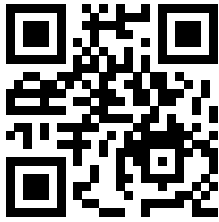
19 décembre 2025

SUJET 1

INSTRUCTIONS (à lire attentivement)

IMPORTANT! Veuillez suivre les instructions suivantes à la lettre sous peine de voir votre examen annulé dans le cas contraire.

1. Vous disposez de deux heures quarante-cinq minutes pour faire cet examen (13h15 – 16h00).
2. Vous devez **écrire à l'encre noire ou bleu foncée**, pas de crayon ni d'autre couleur.
N'utilisez **pas non plus de stylo effaçable** (perte de l'information à la chaleur).
3. Vous avez droit à toute documentation papier.
En revanche, vous ne pouvez pas utiliser d'ordinateur personnel, ni de téléphone portable, ni aucun autre matériel électronique.
4. Répondez aux questions directement sur la donnée, **MAIS** ne mélangez pas les réponses de différentes questions!
Ne joignez aucune feuille supplémentaire ; **seul ce document sera corrigé**.
Si nécessaire, il y a une page de réponse supplémentaire en fin de copie (page 20).
5. Lisez attentivement et *complètement* les questions de façon à ne faire que ce qui vous est demandé. Si l'énoncé ne vous paraît pas clair, ou si vous avez un doute, demandez des précisions à l'un(e) des assistant(e)s.
6. L'examen comporte 5 exercices indépendants sur 20 pages, qui peuvent être traités dans n'importe quel ordre, mais qui ne rapportent pas la même chose (les points sont indiqués, le total est de 125 points).
Tous les exercices comptent pour la note finale.



Question 1 – Échauffement [9 points]

① [5 points] En utilisant RSA, un ami, dont la clé publique est $(2999, 4891)$, vous a transmis le message confidentiel 2954.

Vous avez de votre côté égaré vos clés, mais aviez noté p , q , e et d sur un papier ; vous ne savez cependant plus dans quel ordre. Voici ce que vous retrouvez sur votre papier :

995 43 155 73

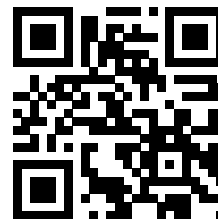
Par contre, vous vous souvenez que vous aviez publié 155.

Comment décidez vous le message reçu de votre ami ?

Exprimez votre réponse sous la forme « $x^y \bmod z$ » et **justifiez** pleinement votre réponse.

Réponse et justification :

Ne pas écrire dans cette zone.



② [4 points] On considère le signal suivant :

$$X(t) = 2 \sin\left(7\pi t + \frac{3\pi}{2}\right) + 3 \sin\left(29\pi t + \frac{\pi}{3}\right) + 4 \sin\left(15\pi t + \frac{5\pi}{6}\right) + 5 \sin\left(20\pi t + \frac{3\pi}{4}\right)$$

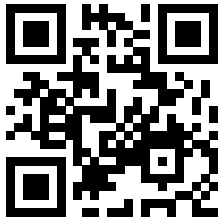
lequel est tout d'abord filtré avec un filtre passe-bas idéal de fréquence de coupure 14 Hz avant d'être échantillonné.

À quelle fréquence au minimum faut-il échantillonner le signal filtré pour pouvoir le reconstruire parfaitement (avec la formule de reconstruction vue en cours) ?

Justifiez pleinement votre réponse.

Réponse et justification :

suite au dos ➞



Question 2 – Tout récursif [32 points]

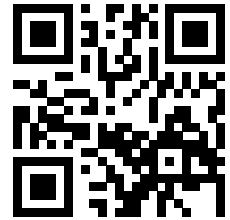
① [5 points] Écrivez un algorithme **somme**, *récursif* et sans boucle qui prend en entrée une liste de nombres entiers et sort la somme de tous ces nombres.

Réponse :

② [4 points] Quelle est la complexité de votre algorithme **somme** écrit en ① ? **Justifiez** votre réponse.

Réponse :

Ne pas écrire dans cette zone.



③ [7 points] On souhaite maintenant utiliser l'algorithme **somme** précédent et un algorithme **Hrec** à définir pour pouvoir calculer l'entropie d'une liste L de comptes de lettres, $L = (n_1, \dots, n_n)$, de la façon suivante :

entropie
entrée : <i>liste L telle que décrite ci-dessus</i>
sortie : <i>entropie des comptes fournis</i>
$n \leftarrow \mathbf{taille}(L)$ $S \leftarrow \mathbf{somme}(L)$ Sortir : $\mathbf{Hrec}(n, S, L)$

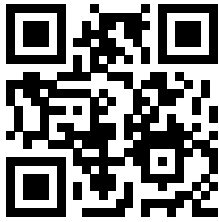
Écrivez l'algorithme **Hrec** de façon *réursive* et sans boucle.

Réponse :

④ [2 points] Quelle est la complexité de votre algorithme **Hrec** écrit en ③ ? **Justifiez** votre réponse.

Réponse :

suite au dos ➞



⑤ [10 points] Écrivez un algorithme **est_trié**, *récuratif* et sans boucle qui prend en entrée une liste de nombres entiers et sort « vrai » si la liste est triée (par ordre croissant ou décroissant, peu importe ; les deux cas doivent fonctionner) et « faux » si la liste n'est pas triée.

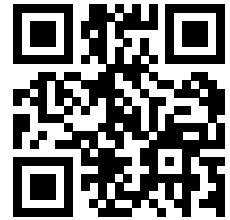
Remarque : testez peut-être votre algorithme avec (4, 5, 5, 4, 3, 2), (2, 3, 4, 5, 5, 4) ou (2, 3, 4, 4, 4, 3, 2).

Réponse :

Ne pas écrire dans cette zone.

⑥ [4 points] Quelle est la complexité de votre algorithme **est_trié** écrit en ⑤ ? **Justifiez** votre réponse.

Réponse :



Question 3 – Réseau [23 points]

Note : (comme toujours, mais particulièrement ici) nous vous conseillons, afin de faire les bons choix, de lire *entièrement* cette question avant de commencer.

Nous souhaitons écrire (des parties d')un programme C++ de simulation de réseau TCP, et plus précisément les nœuds de ce réseau et leurs connexions.

Pour cela, nous avons déjà à disposition les définitions suivantes des types permettant de représenter des adresses IP (type `AdresseIP`) et des tables de routage (type `Table`) :

```
typedef string AdresseIP;

struct Ligne {
    AdresseIP destination;
    AdresseIP proche_voisin;
    unsigned int distance;
};

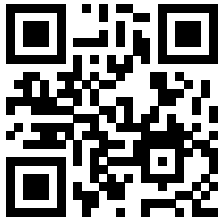
typedef vector<Ligne> Table;
```

① [3 points] En utilisant les types définis ci-dessus, on vous demande de définir ici le type `Noeud` pour représenter un nœud du réseau. Un tel nœud doit contenir :

- son adresse IP ;
- sa table de routage ;
- la liste de ses voisins (*libre* à vous de choisir comment le faire) ;
- un booléen `deja_vu` indiquant si le nœud a déjà été visité ou non ; celui-ci nous servira lors de parcours du réseau (voir la sous-question ② suivante).

Réponse :

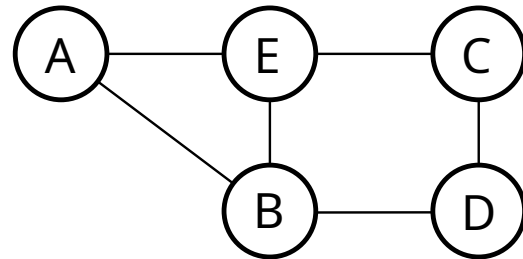
suite au dos ➞



② [9 points] Définissez ci-dessous la fonction `affiche()` qui prend un `Noeud` et une chaîne de caractères et qui affiche (récursivement) tout le réseau à partir de ce nœud.

Par exemple, pour le réseau ci-dessous à droite, l'appel à cette fonction depuis le nœud A donnera :

```
- A connecté à :  
  - B connecté à :  
    - A  
    - D connecté à :  
      - B  
      - C connecté à :  
        - D  
        - E connecté à :  
          - A  
          - B  
          - C  
      - E  
  - E
```

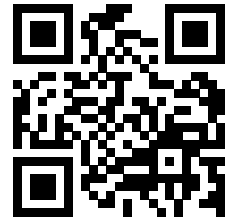


La chaîne de caractères passé en second argument sert à décaler l’affichage de chaque voisin : elle doit avoir une valeur par défaut de "- " et sera augmentée devant de " " (4 espaces) pour chaque nouveau niveau de voisins.

Notez bien que les voisins d’un nœud ne sont affichés que la première fois que ce nœud est affiché (ils ne sont plus affichés si le nœud a déjà été affiché ; utilisez le champ `deja_vu` des nœuds pour cela).

Réponse :

Ne pas écrire dans cette zone.

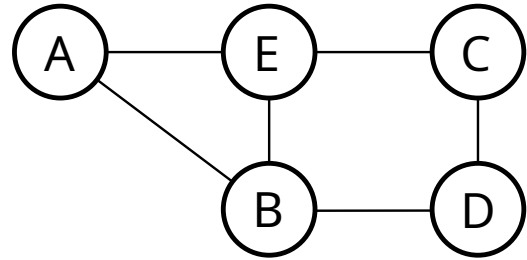


③ [11 points] On vous demande enfin d'écrire la fonction `connecte()` qui prend deux nœuds en argument et qui, *si* les nœuds ne sont pas déjà voisins :

- les connecte (ajoute *chacun* aux voisins de l'autre) ;
- met à jour leur table de routage pour ce nouveau nœud ajouté (s'il existe déjà dans la table, mettre sa distance à 1 et sinon l'ajouter à distance 1).

Par exemple, pour représenter le réseau ci-contre, on aurait écrit :

```
connecte(A, B);  
connecte(A, E);  
connecte(B, D);  
connecte(B, E);  
connecte(C, D);  
connecte(C, E);
```



Réponse :

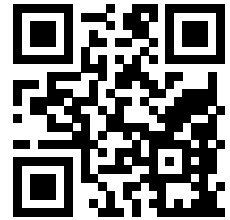
Place supplémentaire pour continuer à répondre au dos 📄



Anonymisation : #0000
p. 10

Question 3

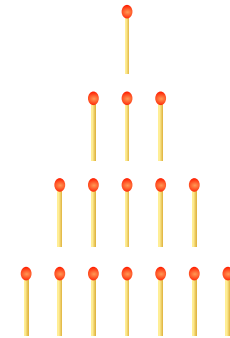
Ne pas écrire dans cette zone.



Question 4 – Retour à Marienbad [29 points]

On s'intéresse ici des parties d'un programme permettant de jouer au jeu de Marienbad, lequel est constitué de quatre lignes ayant respectivement 1, 3, 5 et 7 allumettes (cf image ci-contre).

Les premières questions s'intéressent au codage des situations du jeu, les dernières questions à la sauvegarde sur fichier, en C++, et à un bug possible.



© M0tty, CC BY-SA 3.0, via Wikimedia Commons

① [2.5 points] Pour coder une situation du jeu, si l'on code simplement le nombre d'allumettes de chaque ligne séparément, de combien de bits a-t-on besoin en tout pour coder ?

Justifiez votre réponse.

Réponse et justification :

② [4 points] En utilisant le codage de la question ① précédente, avec en premier (bit(s) de poids fort(s)) le ou les bit(s) représentant la première ligne (0 ou 1 allumette), puis juste après le ou les bits représentant la seconde ligne (0 à 3 allumette(s)), etc., quelle serait la valeur équivalente en décimal (convention non signée) correspondant à la représentation de la situation de jeu 1,2,3,5 ?

Justifiez votre réponse.

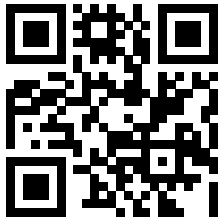
Réponse et justification :

③ [2 points] En utilisant le codage de la question ② précédente, quelle est la situation de jeu correspondant à la valeur décimale 66 ?

Justifiez votre réponse.

Réponse et justification :

suite au dos ➞



④ [1 point] Il y a en fait $2 \times 4 \times 6 \times 8 = 384$ situations de jeu possibles. En utilisant un même nombre de bits pour coder chacune de ces situations, de combien de bits a-t-on besoin au minimum ?

Justifiez votre réponse.

Réponse et justification :

⑤ [4.5 points] Combien de valeurs différentes peut-on coder sur le nombre de bits que vous avez indiqué à la question ④ précédente ?

Combien de valeurs différentes peut-on coder sur le nombre de bits que vous avez utilisé à la question ① ? Expliquez pourquoi il y a un écart entre ce nombre de valeurs et 384 ; p.ex., quelle valeur ne correspond pas à une situation de jeu ?

Retrouver le nombre 384 en comptant le nombre de valeurs qui ne correspondent pas à des situations de jeu dans le codage proposé en questions ① à ③.

Réponses et justifications :

⑥ [10.5 points] Le programme C++ que l'on a écrit nécessite de mémoriser des valeurs dans un tableau `Memory` :

```
typedef array<int, 496> Memory;
```

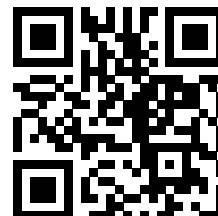
que l'on souhaite pouvoir lire depuis un fichier et aussi sauvegarder dans un fichier.

Écrivez ci-contre deux fonctions `lire()` et `sauvegarder()` qui prennent une `Memory` et un nom de fichier en argument et ne retournent rien.

Ces fonctions devront lancer une/des exception(s) de votre choix en cas d'erreur.

Question 4

Anonymisation : #0000
p. 13



Réponse :

suite au dos ➞



⑦ [4.5 points] Voici un extrait du code produit, lequel est censé faire apprendre à la machine comment mieux jouer :

```
void learn(Memory mem, Play_record record, bool won = true)
{
    Score incr(1);
    if (won) incr = -1;
    for (auto game_rep : record) {
        mem[game_rep] += incr;
        incr = -incr;
    }

    // if last move results from a resign: don't take it into account
    if (record.back() != 0) mem[record.back()] += incr;
}
```

Et voici un extrait de comment cette fonction `learn()` est appelée :

```
int main()
{
    Memory ia_memory{};
    Play_record record;
    bool ia_wins(true);

    //... plein d'autres choses

    learn(ia_memory, record, ia_wins);

    return 0;
}
```

Mais après plusieurs essais, le programme refuse d'apprendre.

Corriger le code ci-dessus pour le programme puisse effectivement apprendre.
Commentez brièvement votre correction ou la/les source(s) d'erreur.

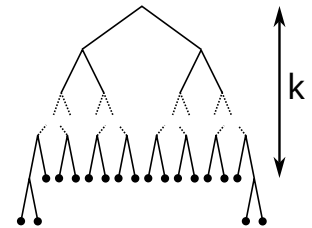
Ne pas écrire dans cette zone.



Question 5 – Huffman et les autres [32 points]

On s'intéresse ici à coder (en binaire) des séquences faites à partir de n lettres différentes (toutes présentes), avec n de la forme $n = 2^k + 2$ pour $k \geq 2$. Par exemple pour $k = 3$ ($n = 10$), on voudrait coder la séquence $Y = \text{ABCDE FGHIJ BBA HHHIII CAGE CFF}$ (sans les blancs).

① [5.5 points] On considère ici le code binaire $\mathcal{C}$ dont l'arbre de codage est constitué (cf image ci-contre) d'un arbre binaire de profondeur k dont deux feuilles ont été chacune remplacée par un sous-arbre binaire de profondeur 1.

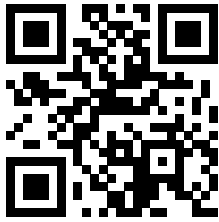


- (a) Est-il toujours possible de construire un tel code (sous les contraintes données plus haut) ?
- (b) Si l'on note p_1, p_2, p_3 et p_4 les quatre probabilités des lettres les plus en bas de l'arbre de codage, quelle est la longueur moyenne d'un tel code ?
- (c) Un tel arbre de codage pourrait-il être un arbre de code de Huffman ? Si oui, à quelle condition (suffisante) ?

Justifiez *chacune* de vos réponses.

Réponses et justifications :

suite au dos ➡

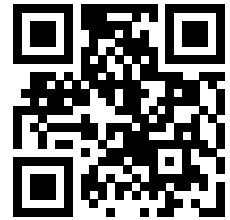


② [5 points] On considère une séquence X telle que celles décrites au début. Soit N la longueur de cette séquence ($N \geq n$). Avec ce qui a été fait jusqu'ici, que pouvez-vous déjà donner comme meilleures bornes (haute et basse) pour l'entropie $H(X)$?

Justifiez votre réponse.

Réponses et justifications :

Ne pas écrire dans cette zone.



③ [8.5 points] Soit n_i le nombre d'apparitions dans X de la i -ème lettre, classées par ordre d'apparition :

$$1 \leq n_1 \leq n_2 \leq n_3 \leq n_4 \leq \dots \leq n_{2^k} \leq n_{2^k+1} \leq n_n$$

On suppose de plus $N \geq 2n - 3$ et $n_4 < n_5$ (inégalité *stricte*).

Quelle est l'entropie minimale pour X (formule en fonction de n et N) ?

Justifiez *pleinement* votre réponse.

Réponse et justification :

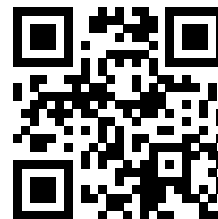
suite au dos ➞



④ [7 points] On suppose maintenant que $N = 3n - 4$ (et toujours $n_4 < n_5$).
Quelle est l'entropie maximale pour X (formule en n uniquement) ? **Justifiez pleinement** votre réponse.

Réponse et justification :

Ne pas écrire dans cette zone.



⑤ **[3 points]** Dans les conditions de la question ④ précédente, quelle est la longueur moyenne du code de Huffman ? (formule en k et n , ou en k seulement) **Indication** : n'oubliez pas la question ①.

Justifiez votre réponse.

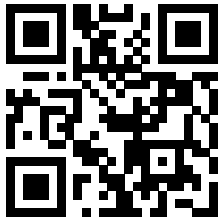
Réponse et justification :

⑥ **[3 points]** À partir de vos réponses aux deux dernières questions, quelle inégalité algébrique pouvez-vous en tirer (une seule suffit ici) ?

Justifiez votre réponse.

Réponse et justification :

Ne pas écrire dans cette zone.



Place supplémentaire pour répondre à n'importe quelle question si nécessaire. Mais
VEUILLEZ INDIQUER LE NUMÉRO DE LA QUESTION TRAITÉE.

Ne pas écrire dans cette zone.