

Information, Calcul et Communication (SMA/SPH) : Correction de l'Examen final

23 décembre 2021

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 (9h15 – 12h00).
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 feuilles supplémentaires ; **seul ce document sera corrigé.**
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 des assistants.
6. L'examen comporte 7 exercices indépendants sur 15 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 120 points).
Tous les exercices comptent pour la note finale.

Question 1 – Florilège [24 points]

① [2 points] Si un nombre entier x s'écrit 0010000000000000 en binaire sur 16 bits, comment s'écrit $\log_2(x)$ en binaire sur 16 bits ?

Réponse : 0000000000001101 (= 13; il y a douze 0 devant).

② [1 point] Sur une machine sur laquelle les `size_t` sont sur 32 bits, quelle est la représentation binaire de la valeur de la variable `s` suivante :

```
const size_t s(-1);
```

Réponse : 111...111 (32 fois que des 1).

③ [2.5 points] Qu'affiche l'extrait de code suivant :

```
double x(3.4);
double y(-9.8);
double z(56.7);

double& t(x);

double* q(&x);

t = z;
q = &y;

cout << "x = " << x << ", y = " << y << ", z = " << z
      << ", t = " << t << ", q ==> " << *q << endl;
```

Réponse : $x = 56.7$, $y = -9.8$, $z = 56.7$, $t = 56.7$, $q ==> -9.8$

④ [3 points] En utilisant RSA, vous souhaitez protéger une information que vous envoyez à un ami dont la clé publique est (25, 95).

Sachant que votre clé publique est (29, 51) et que votre clé privée est 21, vous lui envoyez 23.

Votre ami, dont la clé privée est 49, déchiffre le message qu'il a reçu et trouve 41.

Que pouvez vous dire de la transmission ? Justifiez pleinement votre réponse.

Note : quelques valeurs de $a^b \bmod c$ (classées par a, b, c) :

$23^{21} = 23 \bmod 32$	$23^{21} = 41 \bmod 51$	$23^{25} = 23 \bmod 72$	$23^{25} = 63 \bmod 95$
$23^{29} = 23 \bmod 32$	$23^{29} = 44 \bmod 51$	$23^{49} = 23 \bmod 72$	$23^{49} = 28 \bmod 95$
$41^{21} = 9 \bmod 32$	$41^{21} = 11 \bmod 51$	$41^{25} = 41 \bmod 72$	$41^{25} = 21 \bmod 95$
$41^{29} = 9 \bmod 32$	$41^{29} = 23 \bmod 51$	$41^{49} = 41 \bmod 72$	$41^{49} = 71 \bmod 95$

Réponse :

Si le message de départ était 41 (comme décodé par l'ami), alors nous aurions envoyé $41^{25} \bmod 95$, soit 21 et non pas 23.

Par ailleurs (autre réponse possible), si l'ami avait bien reçu notre message crypté (23), il aurait décodé $23^{49} \bmod 95$, soit 28 et non pas 41.

Il y a donc eu une erreur de transmission.

Note : si l'on suppose que 23 est le message d'origine, en clair, alors il y a de toutes façons un problème puisque l'ami trouve 41!

⑥ [5 points] Quelle est la complexité de l'algorithme ci-dessous. Répondez à droite en justifiant votre réponse.

algo1
entrée : un nombre entier n
sortie : ???
Si $n < 0$ Sortir : algo1 ($-n$) Si $n = 0$ Sortir : algo1 ($n + 1$) Si $n \leq 5$ Sortir : $n - 1$ Si n est pair Sortir : algo1 ($n + 3$) Sortir : algo1 (($n - 1$)/2)

Réponse : $\Theta(\log(n))$ (ou $\Theta(\log(|n|))$) si l'on est plus rigoureux/rigoureux).

Justification simple : si n est pair, on relance sur un impair plus grand, mais la croissance de ces pairs est moins forte (linéaire) que la décroissance (exponentielle) des impairs (division par 2). Donc au final on mettra un nombre d'étapes qui se comporte comme le nombre de divisions par 2, soit logarithmique.

Justification plus complète : si n est pair, on relance sur un impair; et si n est impair, on relance sur $(n - 1)/2$. Le pire cas sera donc lorsque ce nombre lui-même est pair (« ralentit l'algorithme ») :

$$n = 2k + 1 \rightarrow k(\text{pair}) \rightarrow k + 3 \rightarrow (k + 3 - 1)/2 = k/2 + 1 \rightarrow \dots$$

On voit qu'il y a alors des schémas récursifs de nombres comme p.ex. si n est de la forme $2^m + 5$ ou de la forme $2^m + 1$.

Dans ces cas, le nombre d'appels est bien en $\Theta(\log(n))$.

⑥ [5.5 points] Considérons un texte composé uniquement d'espaces et des 26 lettres usuelles, soit en majuscules, soit minuscules (les deux sont possibles dans un même texte), dont l'entropie (telle que définie au cours) est de 2.6 bit.

a) [1.5 points] Est-ce possible? Justifiez brièvement votre réponse.

Réponse : Oui c'est possible : $0 \leq 2.6 \leq \log_2(53)$.

($\log_2(53)$ est compris entre 5 ($\log_2(32)$) et 6 ($\log_2(64)$).)

Note : même si l'on ne considère pas les espaces : $0 \leq 2.6 \leq \log_2(52)$.

b) [4 points] Ce texte enregistré de façon directe/simple (c.-à-d. avec un nombre constant et minimal de bits par lettre) utilise 7.1 Mio.

En utilisant un algorithme de compression, on parvient à réduire de moitié la taille de ce fichier.

Que pouvez-vous dire (au maximum) de l'algorithme de compression utilisé?

Réponse : Pour coder les 53 lettres il faut 6 bits. Si on réduit de moitié, c'est que l'on utilise 3 bits en moyenne par lettre; ce qui *semble* possible avec un code (sans-préfixe et sans perte) relativement bon puisque le code de Shannon-Fano ou celui de Huffman auraient sur ce cas une longueur moyenne entre 2.6 et 3.6 bit.

Pour être sûr (et savoir si le code utilisé est optimal), il faudrait appliquer un code de Huffman et comparer les tailles obtenues.

⑦ [5 points] Considérez le code assembleur ci-dessous dont l'entrée est placée dans le registre **r0** et la sortie sera lue dans le registre **r1**. Tous les registres travaillent ici en double.

```

1: charge    r1, 1.0
2: somme     r2, r0, r0
3: somme     r2, r2, 1.0
4: multiplie r1, r1, r0
5: divise   r1, r1, r2
6: somme     r1, r1, 1.0
7: soustrait r0, r0, 1.0
8: cont_pp  0.0, r0, 2
9: somme     r1, r1, r1

```

Approximation de π		
entrée : <i>un nombre entier $n \geq 1$</i> sortie : <i>un nombre décimal</i>		
$x \leftarrow 1$ Répéter <table border="0"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"> $x \leftarrow x \times \frac{n}{2n+1} + 1$ $n \leftarrow n - 1$ </td> <td></td> </tr> </table> tant que $n > 0$ Sortir : $2x$	$x \leftarrow x \times \frac{n}{2n+1} + 1$ $n \leftarrow n - 1$	
$x \leftarrow x \times \frac{n}{2n+1} + 1$ $n \leftarrow n - 1$		

où l'instruction « cont_pp a, b, N » effectue le test « $a < b$ ».

a) [3 points] Ecrivez un algorithme correspondant au programme donné. (Vous pouvez l'écrire à droite du code si vous préférez.)

On peut aussi l'écrire avec une itération, mais notez alors bien la condition imposée à l'entrée n (sinon il manque l'itération initiale, faite de toutes façons) :

Approximation de π		
entrée : <i>un nombre entier $n \geq 1$</i> sortie : <i>un nombre décimal</i>		
$x \leftarrow 1$ Pour i de n à 1 en descendant <table border="0"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"> $x \leftarrow x \times \frac{i}{2i+1} + 1$ </td> <td></td> </tr> </table> Sortir : $2x$	$x \leftarrow x \times \frac{i}{2i+1} + 1$	
$x \leftarrow x \times \frac{i}{2i+1} + 1$		

b) [2 points] Quelle est la sortie si l'entrée (r0) est 2.0. Donnez la réponse à la fois sous forme d'une formule mathématique (correspondant à l'algorithme) et sous forme d'une fraction réduite.

Réponses :

$$2 \times \left(\frac{\frac{2}{5} + 1}{3} + 1 \right) = \frac{44}{15}$$

Question 2 – Signal [13 points]

On considère le signal $X(t)$ suivant :

$$X(t) = 5 \sin\left(8\pi t + \frac{\pi}{6}\right) + 2 \sin\left(3\pi t + \frac{\pi}{3}\right) + 4 \sin\left(5\pi t + \frac{\pi}{4}\right)$$

Soit $Y(t)$ le signal obtenu après filtrage du signal $X(t)$ par un filtre passe-bas idéal de fréquence de coupure $f_c = 3.1$ Hz.

① [4 points] Si l'on échantillonne $Y(t)$ 6 fois par seconde, quel est le signal obtenu après reconstruction (suivant la formule de reconstruction du cours) ?

Justifiez pleinement votre réponse.

Réponse : Les fréquences du signal sont 4 Hz, 1.5 Hz et 2.5 Hz. Le filtre supprime donc la première (4 Hz). Il ne reste donc dans Y que les deux derniers termes :

$$Y(t) = 2 \sin\left(3\pi t + \frac{\pi}{3}\right) + 4 \sin\left(5\pi t + \frac{\pi}{4}\right)$$

dont la bande passante est 2.5 Hz.

Ce signal est échantillonné à 6 Hz, qui est plus grand que deux fois la bande passante, donc ce signal peut être reconstruit exactement.

② [9 points] Soit $Z(t)$ le signal $Z(t) = X(t) - Y(t)$.

Si l'on échantillonne $Z(t)$ 6 fois par seconde (en partant de $t = 0$), quel est le signal obtenu après reconstruction ?

Justifiez pleinement votre réponse et commentez le résultat obtenu.

Réponse : Suite à la réponse précédente, on a donc

$$Z(t) = 5 \sin\left(8\pi t + \frac{\pi}{6}\right)$$

dont la bande passante est 4 Hz.

Ce signal est échantillonné à 6 Hz, qui est inférieur à deux fois la bande passante, donc ce signal ne peut pas être reconstruit exactement. Il faut regarder plus en détails.

On a donc les échantillons suivants :

$$Z\left(\frac{n}{6}\right) = 5 \sin\left(8\pi \frac{n}{6} + \frac{\pi}{6}\right) = 5 \sin\left(\left(8n + 1\right)\frac{\pi}{6}\right) = 5 \sin\left(\left(-4n + 1\right)\frac{\pi}{6} + 2n\pi\right) = -5 \sin\left(\left(4n - 1\right)\frac{\pi}{6}\right)$$

qui serait donc comme si on avait échantillonné le signal

$$Z'(t) = -5 \sin\left(4\pi t - \frac{\pi}{6}\right)$$

lequel a une bande passante de 2 Hz et serait parfaitement construit.

C'est donc ce signal là qui est reconstruit.

Commentaire : c'est « l'effet stroboscopique » (ou « repliement de spectre »)

Question 3 – Comptabilité [22 points]

On s'intéresse ici à écrire un programme de gestion des ventes d'un magasin spécialisé dans la vente de trois catégories de produits : livres, morceaux de musique et logiciels informatiques (softwares).

On vous demande d'écrire ce programme en utilisant le langage C++ et une approche « procédurale » avec le moins de duplication de code possible.

① [3 points] Chaque catégorie de produits contient les informations suivantes : un titre, un nom (p.ex. auteur, système d'exploitation, ...), le prix (unitaire), la quantité en stock et sa catégorie (livre, musique, logiciel).

Définissez ici un type « **Produit** » pour représenter les produits :

Version simple :

```
struct Produit {
    string titre;
    string nom;
    unsigned int qtt;
    double prix;
    char type;
};
```

Version plus avancée avec enum :

```
enum TProduit { LIVRE, MUSIQUE, SOFT };

struct Produit {
    string titre;
    string nom;
    unsigned int qtt;
    double prix;
    TProduit type;
};
```

② [2 points] Écrivez une fonction « **affiche()** » qui prend un produit en argument et affiche les informations correspondantes, ainsi que le prix total résultant (quantité fois prix unitaire) ; par exemple comme :

[S] Wesnoth, Linux, 3 x 210.2 CHF --> 630.6 CHF

(où « [S] » est la catégorie (« logiciel (software) » ici), « Wesnoth » est le titre, « Linux » le nom, « 3 » la quantité et « 210.20 » le prix unitaire.

Vous êtes libres de choisir le format d'affichage de la catégorie du produit.

Version simple :

```
void affiche(Produit const& p)
{
    cout << '[' << p.type << "]" << " " << p.titre << ", " << p.nom << ", "
         << p.qtt << " x " << p.prix << " CHF"
         << " --> " << p.qtt * p.prix << " CHF"
         << endl;
}
```

Version plus avancée :

(au dos)

```

void affiche(TProduit t) {
    switch(t) {
        case LIVRE : cout << "[L]"; break;
        case MUSIQUE: cout << "[M]"; break;
        case SOFT : cout << "[S]"; break;
        default : cout << "[?]"; break;
    }
}

double prix(Produit const& v)
{
    return v.qtt * v.prix;
}

void affiche(Produit const& p)
{
    affiche(p.type);
    cout << " " << p.titre << ", " << p.nom << ", "
        << p.qtt << " x " << p.prix << " CHF"
        << " --> " << prix(p) << " CHF"
        << endl;
}

```

③ [2 points] On s'intéresse maintenant à la représentation d'une commande dans son ensemble, c.-à-d. un *ensemble* de produits.

Définissez ici un nouveau type « **Commande** », servant à représenter un tel ensemble :

```
typedef vector<Produit> Commande;
```

④ [4 points] Définissez une nouvelle fonction « **affiche()** » prenant une « **Commande** » en argument. Cette fonction devra afficher les informations de chaque produit de la commande, puis finir en affichant le total de la commande, p.ex. :

[S] Wesnoth, Linux, 3 x 210.2 CHF --> 630.6 CHF

[L] ExosC++, JCC, 1 x 40 CHF --> 40 CHF

[M] JomsVik, Amon, 1 x 25 CHF --> 25 CHF

TOTAL = 695.6 CHF

Réponse :

```
void affiche(Commande const& commande)
{
    double total(0.0);

    for (auto const& p : commande) {
        affiche(p);
        total += p.qtt * p.prix; // copié-collé !
    }

    cout << "----" << endl << "TOTAL : " << total << " CHF" << endl;
}
```

Le « copié-collé » du calcul du prix devrait inciter à faire une *fonction* (voir version plus avancée à ②).

⑤ [1 point] Créez une fonction « `ajoute()` » permettant d'ajouter un nouveau produit à une « `Commande` ». Cette fonction devra prendre comme arguments : la commande où ajouter, la catégorie du produit, son titre, son nom, le prix unitaire et la quantité.

```
void ajoute(Commande& stock, char type,
            string const& titre, string const& nom,
            unsigned int quantite, double prix)
{
    stock.push_back({ titre, nom, quantite, prix, type });
}
```

⑥ [2 points] On s'intéresse maintenant à intégrer un système de taxes, lesquelles changent en fonction de la catégorie du produit : 1% pour les livres, 2% pour la musique et 4% pour les logiciels.

Créez une fonction « `calcule_taxe()` » qui prend un produit en argument et retourne la valeur de la taxe pour **un** seul produit correspondant (taxe unitaire, p.ex. le « 8.408 » dans la première ligne de l'exemple d'affichage de la sous-question suivante).

Version simple :

```
double calcule_taxe(Produit const& p)
{
    if (p.type == 'L') return p.prix * 0.01;
    if (p.type == 'M') return p.prix * 0.02;
    if (p.type == 'S') return p.prix * 0.04;
    return 0.0;
}
```


Version plus avancée :

```
double constexpr taxe_livre(0.01);
double constexpr taxe_musique(0.02);
double constexpr taxe_soft(0.04);
// mieux mais encore plus avancé :
// on pourrait aussi ici faire un tableau de const double
// indexé suivant les TProduit :
// const vector<double> taxes{ 0.01, 0.02, 0.04 };

double calcule_taxe(Produit const& p)
{
    switch(p.type) {
        case LIVRE : return p.prix * taxe_livre;
        case MUSIQUE: return p.prix * taxe_musique;
        case SOFT   : return p.prix * taxe_soft;
    }
    return 0.0;

    // la solution encore plus avancée avec un tableau est plus simple ici :
    // return v.prix * taxe[v.type];
}
```

⑦ [8 points] Expliquez ici comment vous modifieriez votre programme écrit jusqu'ici de sorte que l'on intègre les taxes dans le calcul du prix et que l'affichage d'une commande produise, par exemple, le résultat suivant :

[S] Wesnoth, Linux, 3 x (210.2 CHF HT + 8.408) --> 655.824 CHF TTC

[L] ExosC++, JCC, 1 x (40 CHF HT + 0.4) --> 40.4 CHF TTC

[M] JomsVik, Amon, 1 x (25 CHF HT + 0.5) --> 25.5 CHF TTC

TOTAL : 721.724 CHF TTC

où 8.408, 0.4 et 0.5 sont les taxes respectives sur chacun des produits listés.

Il faut modifier :

- le calcul du prix total (mieux en faire une fonction si pas déjà fait ; ou modifier la fonction déjà faite de calcul du prix d'un produit) :
ajouter le calcul de la taxe :
`p.qtt * (p.prix + calcule_taxe(p));`
- modifications affichage d'un produit (`affiche(Produit)`) :
 - parenthèses, « HT » et « TTC »
 - affichage de la taxe
 - affichage du prix taxé/total
- modifications affichage d'une commande (`affiche(Commande)`) :
uniquement ajouter le « TTC », si on a fait une fonction pour le calcul du prix !
(cela doit faire prendre conscience du « copié-collé » sinon : on a **2** choses à modifier (le prix total dans l'affichage du produit et dans l'affichage d'une commande)).

Question 4 – La musique adoucit les mœurs [10 points]

On s'intéresse à l'air musique composé des notes suivantes :

SOL SOL LA SOL DO SI SOL SOL LA SOL RÉ DO

(l'espace n'est là que donner une indication du rythme).

Pour simplifier, on suppose que ces notes sont des sinusoïdes pures :

SOL ($f_1 = 392$ Hz), LA ($f_2 = 440$ Hz), SI ($f_3 = 494$ Hz), DO ($f_4 = 523.3$ Hz) et RÉ ($f_5 = 587.3$ Hz).

① [2.5 points] Si on passe cette mélodie à travers un filtre passe-bas idéal de fréquence de coupure 650 Hz, va-t-on entendre des dissonances à la sortie du filtre? Va-t-on entendre la même mélodie? Justifiez vos réponses.

Réponse :

- un filtre ne crée pas de dissonances en soit, au pire il coupe des notes
- la plus haute des fréquences est 587.3 Hz
- donc aucune note n'est coupée et on a bien la même mélodie

② [3.5 points] Si on échantillonne la mélodie de départ à une fréquence de 1100 Hz, puis qu'on reconstruit le signal (à l'aide de la formule de reconstruction du cours), combien de notes seront reproduites correctement (parmi les 12 notes qui composent la mélodie)? Justifiez votre réponse.

Réponse :

- tout signal en dessous de $f_e/2$, c.-à-d. en dessous de 550 Hz, sera reconstruit correctement
- donc toutes les notes sauf le RÉ sont reproduites correctement, soit 11 notes au total sur la mélodie en question

③ [4 points] En supposant qu'il n'est pas possible de changer la fréquence d'échantillonnage, proposez une méthode pour éviter les dissonances lors de la reconstruction du signal. Décrivez ce que vous faites, puis commentez le résultat obtenu.

Réponse :

- Il faudrait en premier filtrer le signal pour supprimer le RÉ
- p.ex avec un filtre à fréquence de coupure 540 Hz (un peu en dessous de $f_e/2$)
- ce choix permet aussi de transmettre sans dissonance d'autres mélodies (et est donc meilleur que, p.ex. filtrer à 585 Hz)
- après un tel filtre, le signal échantillonné à 1100 Hz sera parfaitement reconstruit (mais bon... ..il manque une note!).

Question 5 – Un petit algorithme [22 points]

Note : lisez la sous-question ④ (en face) avant de répondre à ②.

Soit L une liste de n nombres entiers positifs triés dans l'ordre croissant ; $n \geq 2$.

On cherche à écrire un algorithme dont l'entrée soit la liste L et la sortie soit un nombre m positif non nul et inférieur strictement à n , tel que (la valeur absolue de) l'écart entre la somme des m premiers nombres de L et la somme des $n - m$ derniers nombres de L soit minimale.

Par exemple, pour $L = (1, 1, 2, 2, 2, 4, 5, 5)$, un tel algorithme sortira 6 puisque la somme des 6 premiers éléments de L est 12 et celle des 2 derniers est 10, dont l'écart est de 2 (en valeur absolue).

① [2 points] À quoi pourrait servir un tel algorithme ?

Réponse : À séparer en deux le tableau des fréquences dans les étapes de l'algorithme de Shannon-Fano.

② [6 points] Écrivez un algorithme pour résoudre ce problème.

Réponse :

Voici un premier algorithme, simple :

Couper en deux
entrée : une liste L [...]
sortie : un rang
$n \leftarrow \text{taille}(L)$
Si $n \leq 2$
Sortir : 1
$m \leftarrow 1$
$d \leftarrow +\infty$
Pour i de 1 à $n - 1$
$s_1 \leftarrow 0$
Pour j de 1 à i
$s_1 \leftarrow s_1 + L[j]$
$s_2 \leftarrow 0$
Pour j de $i + 1$ à n
$s_2 \leftarrow s_2 + L[j]$
$\delta \leftarrow s_1 - s_2 $
Si $\delta \leq d$
$m \leftarrow i$
$d \leftarrow \delta$
Sortir : m

③ [2 points] Quelle est la complexité de votre algorithme proposé en ② ? Justifiez votre réponse.

Réponse : La complexité de l'algorithme précédent est en $\Theta(n^2)$. On parcourt en effet la liste en entier et, pour chaque position de ce parcours (i dans l'algorithme) on reparcourt toute la liste (j) pour calculer les deux sommes.

④ [12 points] Écrivez un algorithme de complexité temporelle $\Theta(n)$ pour résoudre le problème proposé. Si votre réponse à ② est déjà en $\Theta(n)$, vous n'avez rien à faire ici (et serez, bien entendu, noté(e) sur la somme des points des deux sous-questions).

Réponse : Il y a plein de façon d'écrire un algorithme en $\Theta(n)$ pour résoudre cette tâche. En voici quelques uns.

<p>Couper en deux entrée : une liste L [...] sortie : un rang</p> <pre> n ← taille(L) m ← 1 s1 ← L[m] s2 ← L[n] Tant que m < n Si s1 < s2 m ← m + 1 s1 ← s1 + L[m] Sinon n ← n - 1 s2 ← s2 + L[n] Si s1 > s2 Sortir : m Sortir : m + 1 </pre>	<p>Couper en deux entrée : une liste L [...] sortie : un rang</p> <pre> s1 ← 0 Pour tout x ∈ L s1 ← s1 + x m ← 1 s2 ← 0 Tant que s2 < s/2 s2 ← s2 + L[m] m ← m + 1 Si 2 × s2 > s1 + L[m - 1] Sortir : m - 1 Sortir : m </pre>	<p>Couper en deux entrée : une liste L [...] sortie : un rang</p> <pre> n ← taille(L) Si n ≤ 2 Sortir : 1 s ← 0 Pour tout x ∈ L s ← s + x Pour m de 1 à n - 1 Si s ≤ 2 × L[m] Si L[m] < s Sortir : m + 1 Sortir : m s ← s - 2 × L[m] Sortir : n - 1 </pre>
--	--	--

Question 6 – Mal éclairé(e) [11 points]

On considère le programme C++ suivant, qui comporte une erreur de conception (dont nous nous occuperons en sous-questions ② et ③) :

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct Couleur {
6      unsigned int rouge;
7      unsigned int vert;
8      unsigned int bleu;
9  };
10
11 struct Ampoule {
12     const string name;
13     Couleur couleur;
14     bool allumee;
15 };
16
17 void affiche_couleur(Couleur const& co, bool saut_de_ligne = true) {
18     cout << "R : " << co.rouge << ", V : " << co.vert << ", B : " << co.bleu;
19     if (saut_de_ligne) cout << endl;
20 }
21
22 bool valide(Couleur const& co) {
23     constexpr unsigned int max(256);
24     return (co.rouge < max) and (co.vert < max) and (co.bleu < max);
25 }
26
27 void affiche_etat_ampoule(const Ampoule* ptr)
28 {
29     if (ptr == nullptr) {
30         cout << "L'ampoule a claqué !" << endl;
31         return;
32     }
33
34     cout << "L'ampoule " << ptr->name << " est ";
35     if (not ptr->allumee ) {
36         cout << "éteinte." << endl;
37     } else {
38         cout << "allumée ; sa couleur est : ";
39         affiche_couleur(ptr->couleur);
40     }
41 }
42
43 // Suite du programme en face --->
```

```

44 // Suite du programme de la page précédente
45 Ampoule* cree_ampoule(string const& name, Couleur const& couleur, bool allume,
46                       bool informe = false)
47 {
48     if (name.empty()) {
49         if (informe) cout << "Erreur : nom vide pour l'ampoule" << endl;
50         return nullptr;
51     }
52     if (not valide(couleur)) {
53         if (informe) {
54             cout << "Erreur : mauvaise couleur (";
55             affiche_couleur(couleur, false);
56             cout << ") pour l'ampoule \"" << name << "' " << endl;
57         }
58         return nullptr;
59     }
60
61     Ampoule ampoule({name, couleur, allume});
62
63     if (informe) {
64         cout << "Initialisation de l'ampoule : ";
65         affiche_etat_ampoule(&ampoule);
66     }
67
68     return &ampoule;
69 }
70
71 int main() {
72     const Ampoule ampoule1({ "de la cuisine"s, {255, 0, 125}, true });
73     affiche_etat_ampoule(&ampoule1);
74
75     const Ampoule* p_ampoule2(cree_ampoule("de la chambre"s, {255, 255, 999},
76                                           true, true));
77     affiche_etat_ampoule(p_ampoule2);
78
79     const Ampoule* p_ampoule3(cree_ampoule("du salon"s, {255, 0, 125}, false));
80     affiche_etat_ampoule(p_ampoule3);
81
82     return 0;
83 }

```

① [4 points] Nous allons d'abord examiner l'exécution du code comme si le programme fonctionnait correctement et que le problème de conception n'avait aucun impact.

Quel serait l'affichage produit par le programme si c'était le cas (c.-à-d. quel est l'affichage attendu par l'auteur(e) qui n'a pas vu sa faute)?

Réponse :

L'ampoule de la cuisine est allumée ; sa couleur est : R : 255, V : 0, B : 125

Erreur : mauvaise couleur (R : 255, V : 255, B : 999) pour l'ampoule "de la chambre"

L'ampoule a claqué !

L'ampoule du salon est éteinte.

Attention à la valeur par défaut (*false*) du dernier argument de *cree_ampoule* et attention aussi au fait que le « bleu » de l'ampoule de la chambre n'est pas valide.

② [3 points] L'auteur(e) de ce code ne prête aucune attention aux avertissements (« *warning* ») du compilateur (et compile d'ailleurs sans option). C'est bien dommage, car en fait son programme compile, mais ne se comporte pas comme attendu (et le « *warning* » aurait pu l'avertir!).

Quel est le problème (expliquez le en une phrase) et quel est, selon vous, son impact sur le `main()` ? (Y-a-t-il une modification d'affichage par rapport à ① ?)

Réponse : Le problème est en ligne 68 de retourner l'adresse d'une variable locale.

L'impact sur le `main()` est difficile à prévoir (« *undefined behaviour* »), mais certainement un impact (« *Segmentation Fault* » ?) sur l'affichage de l'ampoule du salon.

③ [4 points] Proposez une correction pour que le programme fonctionne comme attendu.

Expliquez/Justifiez votre choix en détails.

Réponse : (réponse simple) Si l'on tient à garder les pointeurs, il faudrait faire de l'allocation dynamique (`new`). Et ne pas oublier de `delete` dans le `main()`.

(réponse plus correcte) Mais en fait il n'y a pas lieu du tout d'utiliser les pointeurs dans ce programme. On pourrait tout à fait retourner une `Ampoule` en tant que telle (et donc retourner (la copie de) la variable locale elle-même).

Question 7 – Pesées [18 points]

On considère les résultats de pesées sur une balance indiqués par D si la balance a fortement penché à droite, d si elle a légèrement penché à droite, G si elle a fortement penché à gauche, g si elle a légèrement penché à gauche, et E si elle est restée à l'équilibre.

Le résultat suivant nous a été transmis (15 lettres) :

DGGd*****Eg

où * indique qu'une lettre a été effacée (il y a donc 9 résultats perdus/effacés, qui pourraient chacun être n'importe quelle lettre parmi D, d, G, g ou E).

① [4 points] Quelle est l'entropie minimale (telle que définie en cours) pour le résultat qui nous a été transmis ? Exprimez votre réponse sous forme de résultat numérique incluant éventuellement des \log_2 , et **justifiez** votre réponse.

Réponse : L'entropie est minimale pour la situation la plus répétitive, ce qui est le cas ici en répétant la lettre déjà la plus fréquente, soit le G : DGGdGGGGGGGGEG

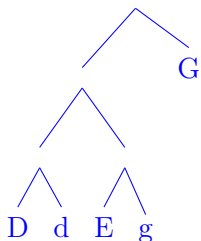
On a donc les fréquences : $\frac{11}{15}$ pour G et $\frac{4}{15}$ pour les quatre autres lettre.

Soit une entropie de :

$$H_{\min} = \frac{11}{15} \log_2 \left(\frac{15}{11} \right) + \frac{4}{15} \log_2(15) = \log_2(15) - \frac{11}{15} \log_2(11)$$

② [4 points] Proposez un code de Huffman (binaire) pour la situation considérée dans la sous-question précédente (①). Quelle est sa longueur moyenne ?

Réponse :



Par exemple :

G : 0,
D : 111,
d : 110,
E : 101,
g : 100

$$L(C) = \frac{11}{15} + 4 \times 3 \times \frac{1}{15} = \frac{23}{15}$$

③ [3 points] Quelle est l'entropie maximale (telle que définie en cours) pour le résultat qui nous a été transmis ? Exprimez votre réponse sous forme de résultat numérique incluant éventuellement des \log_2 , et **justifiez** votre réponse.

Réponse : L'entropie est maximale pour la situation la plus équiprobable, ce qui est le cas ici en répétant chaque lettre trois fois : DGGdDDddFEEggEg

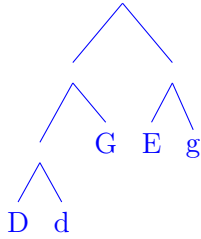
Dans ce cas, on sait que l'entropie est $\log_2(5)$ bit.

④ [4 points] Proposez un code de Huffman (binaire) pour la situation considérée dans la sous-question précédente (③). Quelle est sa longueur moyenne ?

Réponse :

(au dos)

Par exemple :



Par exemple :

G : 10,
D : 111,
d : 110,
E : 01,
g : 00

$$L(C) = \frac{1}{5} (3 \times 2 + 2 \times 3) = \frac{12}{5} = 2.4$$

© [3 points] Commentez les résultats obtenus. (Note : $\log_2 3 \simeq 1.58$, $\log_2 5 \simeq 2.32$, $\log_2 11 \simeq 3.46$.)

Réponse : On sait qu'on doit avoir l'entropie inférieure ou égale à la longueur moyenne du code de Huffman ; ce qui est bien le cas dans les deux cas :

$$H_{\min} \simeq 1.58 + 2.32 - 11 \times 3.46/15 \simeq 3.9 - \frac{38.06}{15} \simeq 1.9 - \frac{8.06}{15} \simeq 1.4 < \frac{23}{15} \simeq 1.5333$$

et

$$\log_2 5 \simeq 2.32 < \frac{12}{5} = 2.4$$