



Concours national d'informatique

**Épreuve écrite d'algorithmique**  
**Louvain-la-Neuve**

Samedi 3 mars 2012

MRKRPXZKRMTFRZ



Extrait de *Tintin au Tibet*, par Hergé

## 1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

### Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien, ou préparez votre présentation pour l'entretien.
- N'oubliez pas de passer une bonne journée.

### Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe, sinon ça va barder.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

## 2 Sujet

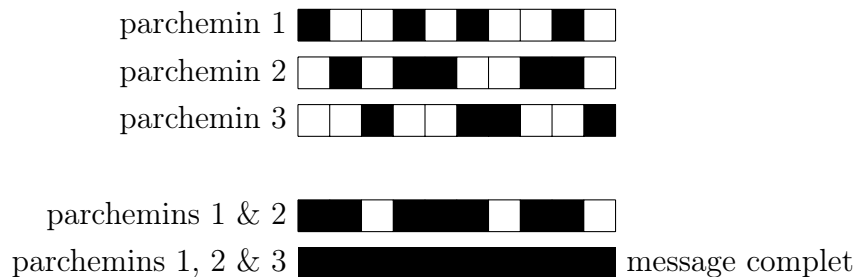
### Introduction

*Trois frères unys. Trois Licornes de  
conserve vogant au Soleil de midi  
parleron.  
Car c'est de la lumière que  
viendra la lumière. Et resplendira  
20 37 42 N. 70<sup>1</sup> 52 15 W.  
la † de l'Aigle.*

Nous sommes en 1698. *La Licorne*, un fier vaisseau de troisième rang de la flotte de Louis XIV, fait voile pour l'Europe. Mais voilà qu'apparaît un navire de flibustiers, qui gagnent de vitesse *La Licorne* et, avec de féroces hurlements, montent à l'abordage. Leur chef, Rackham le Rouge, s'empare du navire. Mais le chevalier de Hadoque parvient à s'enfuir et fait sauter son vaisseau.

À la dernière page des *Mémoires de François Chevallier de Hadoque* figure une sorte de testament dans lequel il déclare léguer à chacun de ses  $n$  fils un modèle, construit et gréé par lui du vaisseau qu'il avait fait sauter jadis pour ne pas le laisser aux mains des pirates. Et, détail amusant, il demande à ses fils de déplacer légèrement vers l'arrière le grand mât de chacun de ces modèles. « *Ainsi* », ajoute-t-il, « *la vérité sera complète.* »

En effet, chaque *Licorne* renferme un parchemin<sup>2</sup>, contenant une partie du message sur  $b$  bits, chaque bit étant dévoilé<sup>3</sup> ou manquant. Lorsqu'on superpose les parchemins et qu'on les porte à la lumière, la transparence révèle tous les bits dévoilés par au moins un parchemin.



Les parchemins considérés aux questions 1 à 4 feront parfois 32 bits, parfois 1 000 bits. Dans ce dernier cas, par commodité, les bits dévoilés seront consécutifs et connexes, c'est-à-dire que chaque parchemin de 1 000 bits pourra être représenté par son intervalle de bits dévoilés, par exemple [42,168] pour « tous les bits du 42<sup>e</sup> au 168<sup>e</sup> ».

### Partie 1 – Calques.

#### Question 1

(2 points)

Indiquez les structures de données que vous allez utiliser pour représenter les messages. Cette question est importante car vous utiliserez vos structures pour écrire les fonctions des questions suivantes. Relisez donc entièrement le sujet ainsi que les questions qui suivent avant de répondre.

---

1. Septante, c'est ça ?  
2. Désolé pour le spoil.  
3. Dévoilé signifie qu'on connaît la valeur de bit, qu'elle soit 1 ou 0. On attachera plus d'importance au fait de recouvrer le message entier qu'au message lui-même.

## Question 2

(3 points)

« Je suis persuadé que si nous parvenons à réunir les  $n$  parchemins, nous découvrirons les diamants de Rackham le Rouge. »

Écrivez une fonction qui prend en argument une liste de parchemins sur 32 bits et détermine s'ils suffisent à recouvrir tout le message. Même question avec 1 000 bits.

## Question 3

(2 points)

Sur les  $n$  parchemins, en existe-t-il deux qui dévoilent un même bit du message ? Écrivez une fonction qui répond à cette question pour 32 bits, puis pour 1 000 bits.

## Question 4

(3 points)

Écrivez une fonction qui détermine si tous les parchemins sont nécessaires à l'obtention du message complet (ou si au moins l'un d'eux peut être perdu sans danger). Encore une fois<sup>4</sup>, il faudra répondre à cette question pour 32 bits, puis pour 1 000 bits.

## Question 5

(3 points)

On considère l'algorithme suivant appliqué aux parchemins :

Tant qu'il manque un bit du message

Ajouter à une liste `mesParchemins` le premier parchemin qui dévoile ce bit.

Donner une liste de parchemins de 4 bits pour lequel cet algorithme ne retourne pas la solution faisant intervenir le moins de parchemins possible.

## Question 6

(4 points)

Partant d'un ensemble de parchemins, comment en déterminer le nombre minimum suffisant à recouvrir tout le message ? Quelle est la complexité de votre algorithme ?

## Partie 2 – Polynômes.

À présent, le chevalier de Hadoque souhaite que  $p$  de ses descendants puissent reconstituer le message ( $p < n$ ), quels qu'ils soient, mais que  $p - 1$  ne le puissent pas, quels qu'ils soient.

Pour ce faire, il s'appuie sur le théorème suivant :

**Théorème.** Il existe une unique fonction de la forme  $f(x) = a_{p-1}x^{p-1} + a_{p-2}x^{p-2} + \dots + a_2x^2 + a_1x + a_0$  passant par  $p$  points  $(1, f(1)), \dots, (p, f(p))$ .

Tout d'abord, il choisit un nombre secret<sup>5</sup>  $a_0$ . Ensuite, il détermine au hasard les coefficients  $a_{p-1}, \dots, a_1$  de  $f$ , puis calcule  $f(1), \dots, f(n)$  pour pouvoir écrire  $(1, f(1))$  sur le premier parchemin,  $(2, f(2))$  sur le deuxième, et ainsi de suite.

---

4. Je n'ai même pas fait exprès.

5.  $a_0$  peut représenter par exemple la concaténation de la latitude et de la longitude du trésor enfoui.

**Exemple.** Le chevalier de Hadoque cherche à cacher le nombre 42 à ses 5 enfants, et souhaite que 3 d'entre eux puissent le retrouver ( $n = 5, p = 3$ ). Il choisit par exemple la fonction  $f(x) = x^2 - 13x + 42$ . On a :

$x$	1	2	3	4	5
$f(x)$	30	20	12	6	2

Il écrit alors (1, 30) sur le 1<sup>er</sup> parchemin, (2, 20) sur le 2<sup>e</sup> puis (3, 12), (4, 6) et enfin (5, 2). D'après le théorème, trois parchemins suffisent pour retrouver l'expression de  $f(x)$ , et donc le secret 42.

### Question 7 (3 points)

Écrire une fonction qui prend en argument les coefficients  $a_{p-1}, \dots, a_0$  et un entier  $k$  et retourne  $f(k)$ .

### Question 8 (4 points)

Aujourd'hui,  $p$  descendants se retrouvent pour accéder au trésor. Ils possèdent les parchemins  $(k_1, f(k_1)), \dots, (k_p, f(k_p))$ .

On dispose de fonctions  $\ell_1, \dots, \ell_p$  telles que :

$$\ell_j(k_i) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon.} \end{cases}$$

Par exemple,  $\ell_3(k_1) = \ell_3(k_2) = \ell_3(k_4) = 0$  mais  $\ell_3(k_3) = 1$ .

Exprimer en fonction des fonctions  $\ell_j$  l'unique fonction  $f$  passant par tous les points  $(k_1, f(k_1)), \dots, (k_p, f(k_p))$ . En déduire le nombre secret.

Vous ne pouvez attaquer les questions suivantes que si vous avez trouvé la voie.

### Question bonus 9 (3 points)

Donnez un exemple simple de fonction  $\ell_j$  qui ait la propriété définie à la question 8.

### Question bonus 10 (5 points)

Comment concevoir les parchemins de la partie 1 de sorte que  $p$  descendants puissent reconstituer le secret quels qu'ils soient et que  $p - 1$  ne le puissent pas? En particulier, quelle doit être la taille minimale de ces parchemins pour que ce soit possible?

### Question bonus 11 (1 point)

Quel est le nom du premier journal à avoir publié les aventures de Tintin <sup>6</sup>?

Le sujet est sur 24 points, et les questions bonus rapportent au total 9 points, plus 1 point de présentation.

---

6. Indice : regardez attentivement l'affiche Prologin de cette année.