

---

Olympiades Régionales  
d'Informatique Franco-Australiennes  
13 mars 2011

---

Durée : 4 heures

4 problèmes

# Problème 1

## Étagère

**Limites de temps et de mémoire : 1 seconde, 32 MO**

Dans la ville de Harigato se trouve une bibliothèque qui contient une collection de livres sur l'ancien ordre des moines Harigatan. Ces livres sont stockés côte à côte en une longue ligne, et n'ont pas tous la même hauteur. Certains de ces livres ont des milliers d'années et sont irremplaçables.

Malheureusement, une poussée de moisissure a commencé à se former dans la collection, détruisant les livres contaminés. Les infections de moisissure commencent dans un ou plusieurs livres contaminés. Si un livre plus petit se trouve à côté d'un livre contaminé, il ne risque pas d'être atteint par la moisissure. Cependant, les livres de même hauteur ou plus grands qu'un livre adjacent contaminé seront également contaminés.

La bibliothèque aimerait déterminer comment de telles poussées affecteraient leur collection, et font appel à vous pour les aider. Étant donné l'ordre et les hauteurs des livres sur l'étagère, quels livres sont irremplaçables et quels livres sont contaminés au départ, votre objectif est de déterminer combien de livres irremplaçables vont être détruits.

### Entrée

La première ligne de l'entrée contiendra trois entiers positifs,  $N M I$ , représentant respectivement le nombre total de livres sur l'étagère, le nombre de livres contaminés au départ et le nombre de livres irremplaçables. Il y aura au maximum 100 000 livres sur l'étagère.

La deuxième ligne de l'entrée contiendra  $N$  entiers positifs, décrivant la hauteur de chacun des  $N$  livres. Les hauteurs sont toutes inférieures ou égales à 1 000 000. Les livres sont posés sur l'étagère dans l'ordre donné et sont numérotés de 1 à  $N$ .

La troisième ligne de l'entrée contiendra  $M$  entiers, décrivant quels livres ont été contaminés par la moisissure. Chacun de ces entiers sera compris entre 1 et  $N$ . Aucun livre n'apparaîtra plus d'une fois dans cette liste.

La quatrième et dernière ligne de l'entrée contiendra  $I$  entiers, indiquant quels livres sont irremplaçables. Chacun de ces entiers sera entre 1 et  $N$ . Aucun entier n'apparaîtra plus d'une fois dans cette liste.

### Sortie

Votre sortie doit consister en un unique entier – le nombre de livres irremplaçables qui seraient contaminés par la moisissure dans le scénario donné.

### Exemple d'entrée

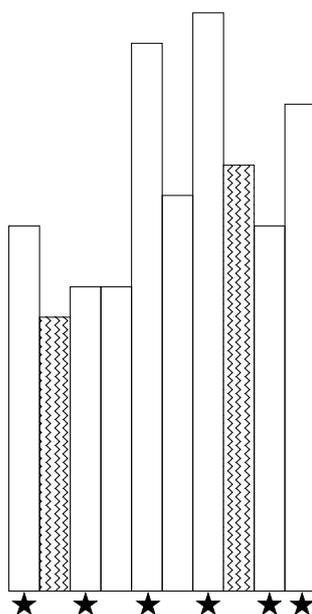
10 2 6  
 12 9 10 10 18 13 19 14 12 16  
 2 8  
 1 3 5 7 9 10

### Exemple de sortie

4

### Explication

L'exemple d'entrée correspond à l'étagère illustrée ci-dessous. Les livres contaminés par la moisissure sont grisés. Les livres irremplaçables sont marqués d'une étoile. La moisissure du livre 2 peut se propager vers les livres irremplaçables 1, 3 et 5. La moisissure du livre 8 peut se propager uniquement vers le livre irremplaçable 7. Ainsi, cela donne un total de 4 livres irremplaçables contaminés par la moisissure une fois que l'infection s'est propagée.



### Score

Le score pour chaque test d'entrée sera de 100 % si la bonne réponse est écrite sur la sortie, ou de 0 % sinon.

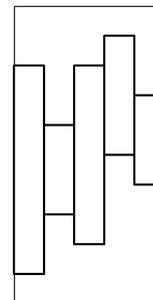
## Problème 2

### La maison aux mille lames

**Limites de temps et de mémoire : 1 seconde, 32 MO**

Seiko étudie le *katana* (épée). Elle s'entraîne du matin au soir sous l'oeil attentif de son maître, un vieil homme qui lui fait profiter de sa sagesse cachée sous la forme de messages cryptiques.

Aujourd'hui, son maître l'emmène au bord du lac *koi* et lui présente une planche de bois plate. Il peint consciencieusement une série de bandes verticales de même largeur sur la planche. Le bord droit de chaque bande touche le bord gauche de la suivante, le dessin complet formant ainsi une forme contiguë qui couvre toute la largeur de la planche. Un exemple d'un tel chef d'oeuvre est présenté sur la droite. Le maître explique alors la leçon du jour à Seiko :



“Élégance. Le maître katana ne gâche ni son temps ni ses efforts. Chaque mouvement de ta lame doit être précis et plein de sens. Prend ton épée et découpe la forme que j'ai peinte sur cette planche, afin de l'extraire en un seul morceau.”

Seiko est précise dans ses mouvements, et elle peut découper un segment de droite quelconque en une seule coupe, quel que soit l'angle du segment, que ses extrémités aient déjà été découpées ou non.

Après avoir médité sur les mots de son maître, Seiko comprend que son objectif est maintenant de déterminer le plus petit nombre de coupes qu'elle doit effectuer pour extraire la forme peinte.

#### Entrée

La première ligne consiste en deux entiers séparés par un espace,  $W$  et  $H$  ( $1 \leq W, H \leq 100\,000$ ), représentant respectivement la largeur et la hauteur de la planche de bois. La planche peut être vue comme un rectangle avec des coordonnées allant de  $(0, 0)$  à  $(W, H)$ .

La deuxième ligne consiste en  $W$  entiers décrivant les bords du haut des  $W$  rectangles peints. Chaque hauteur sera entre 1 et  $H$  inclus.

La troisième ligne consiste en  $W$  entiers décrivant les bords du bas des  $W$  rectangles peints. Chaque hauteur sera entre 0 et  $H - 1$  inclus. On vous garantit que la hauteur du bord du haut de chaque rectangle peint est strictement supérieure à la hauteur de son bord du bas, et que l'ensemble des rectangles peints constitue une forme connectée, c'est à dire que pour tout  $i > 1$ ,  $bas_i < haut_{i-1}$  et  $bas_{i-1} < haut_i$ .

Pour 40% des points disponibles,  $W \leq 1\,000$ .

#### Sortie

Vous devez écrire une ligne contenant un entier sur la sortie – le nombre minimum de coupes nécessaires pour extraire la forme désirée de la planche.

### Exemple d'entrée

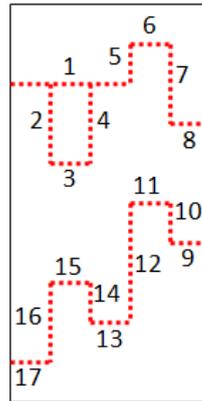
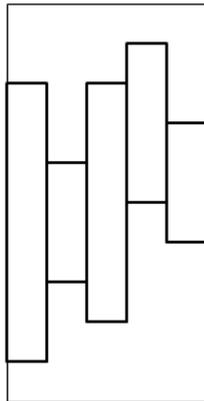
5 10  
 8 6 8 9 7  
 1 3 2 5 4

### Exemple de sortie

17

### Explication

L'exemple d'entrée correspond à la planche illustrée ci-dessous à gauche. Les 17 coupes sont illustrées à droite.



### Score

Le score pour chaque test d'entrée sera de 100 % si la bonne réponse est écrite sur la sortie, ou de 0 % sinon.

## Problème 3

### Vie

**Fichier d'entrée :** *life.k.in*

**Fichier de sortie :** *life.k.out*

#### Tâche à sortie uniquement

Aujourd'hui est le jour le plus important de votre carrière d'exobiologiste. Après de nombreuses tentatives, vous avez enfin réussi à maintenir un micro trou de ver stable pendant plus de 20 nanosecondes entre votre laboratoire et PHR 5426b, l'exoplanète la plus similaire à la terre qui ait été découverte jusqu'à ce jour. Vous avez même pu reproduire l'expérience 10 fois avant que le plasma de votre hypergénérateur ne s'effondre. 20 nanosecondes est juste assez de temps pour que suffisamment de photons traversent le trou de ver et soient capturés par la caméra de votre xPhone 5.

Utiliser un xPhone 5 à 49 euros pour capturer les images les plus importantes du siècle peut sembler bizarre, mais le capteur xxd de sa caméra est de loin le plus sensible jamais fabriqué, et même bien meilleur que les capteurs à 500 000 euros créés pour Hubble 5.

Cependant, après avoir annoncé la grande découverte au monde et convoqué la plus grande conférence de presse depuis la sortie de l'xPhone 5 lui-même, vous réalisez que vous avez un problème : comme le trou de ver est vraiment tout petit, vous avez dû prendre de nombreuses photos pour obtenir une image complète de chaque forme de vie. Malheureusement, vous étiez si enthousiaste suite à votre découverte que vous avez oublié de noter quelles photos correspondent à quelles parties de l'image complète. Vous devez absolument les réordonner avant que la conférence de presse ne commence.

#### Entrée

On vous donne dix fichiers d'entrée *life.k.in* ( $1 \leq k \leq 10$ ), décrivant les dix images mélangées.

La première ligne de chaque fichier contient quatre entiers,  $W H C R$ , indiquant que l'image est composée d'une grille de  $C \times R$  blocs ( $C$  colonnes et  $R$  rangées), chaque bloc étant une petite image de  $W \times H$  pixels ( $W$  colonnes et  $H$  rangées).

Suivent  $C \times R$  groupes de  $H$  lignes. Chaque groupe décrit un bloc par  $H$  lignes de  $W$  entiers compris entre 0 et 255 inclus, donnant la luminosité des pixels composant le bloc. Les blocs sont numérotés de 1 à  $C \times R$  et sont donnés dans cet ordre. Votre seul problème est que cet ordre est sans le moindre rapport avec la position de chaque bloc dans l'image d'origine.

Votre objectif est de déterminer dans quel ordre les  $C \times R$  blocs doivent être placés pour former une image aussi proche que possible que l'image d'origine. Comme vous ne disposez pas des images d'origine, vous ne pouvez compter que sur le fait qu'il s'agit d'images de formes de vie sur une planète lointaine.

#### Sortie

Pour chaque fichier d'entrée *life.k.in*, vous devez produire un fichier de sortie *life.k.out* qui contient votre solution.

Chaque fichier solution doit contenir  $R$  lignes de  $C$  entiers tous distincts, chacun étant le numéro d'un bloc. Ces nombres doivent être placés de telle manière que répartir les blocs correspondants de cette manière sur une grille  $C \times R$  donne une image qui ressemble autant que possible à une forme de vie extra-terrestre.

### Exemple d'entrée

```

2 2 3 2
0 128
0 0
0 0
0 128
0 0
128 0
128 0
0 0
0 0
128 128
128 128
0 0
    
```

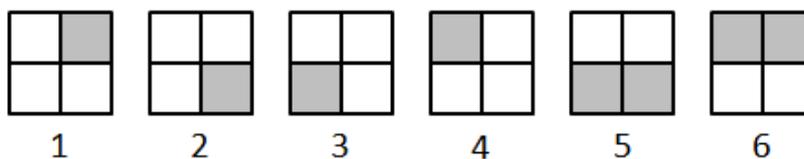
### Exemple de sortie

```

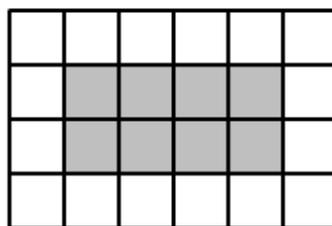
5 1 2
6 4 3
    
```

### Détails de l'entrée

L'entrée décrit 6 blocs, comme illustré ci-dessous :



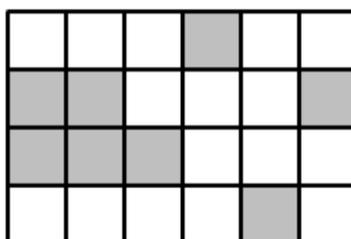
L'image d'origine de cette forme de vie très simple est illustrée ci-dessous :



L'exemple ci-dessus n'est pas représentatif des fichiers d'entrée fournis.

### Détails de la sortie

La sortie décrit un ordre des blocs qui correspond à l'image recomposée ci-dessous :



## Score

Votre score pour chaque fichier d'entrée sera déterminé comme suit :

- Si votre solution ne contient pas exactement  $R$  lignes de  $C$  entiers tous distincts et entre 1 et  $R \times C$ , elle obtiendra un score de 0%.
- Sinon, soit  $A$  le nombre total de paires adjacentes (partageant un côté) de blocs au sein de l'image ( $A = C \times R \times 2 - C - R$ ), et soit  $B$  le nombre de paires de blocs adjacents dans votre sortie qui sont adjacents de la même manière (mêmes côtés) dans l'image d'origine. Votre solution obtiendra un score de  $100 \times B/A$  arrondi à l'entier inférieur (exprimant un pourcentage de points).

Par exemple, dans l'exemple ci-dessus, le fichier de sortie parfait serait :

```
2 5 3
1 6 4
```

Comme ceci correspond exactement à l'image d'origine, les 7 paires de blocs adjacents sont correctes, donc cette sortie aurait obtenu un score de 100% ( $100 \times 7/7$ ).

Dans l'exemple de sortie cependant, seules les paires (5, 6) et (6, 4) sont adjacentes de la même manière que dans l'image d'origine. Cette sortie obtiendrait donc un score de 28% ( $100 \times 2/7 = 28$ ).

## Experimentation

Utilisez la section “trial runs” du site de soumission pour expérimenter avec les données d'entrée. Vous pourrez y soumettre des sorties valides, et visualiser l'image recomposée correspondante.

## Soumission pour un sujet à sortie uniquement

Pour soumettre vos fichiers de sortie, vous devez faire ce qui suit :

- Créer un fichier zip contenant tous vos fichiers de sortie. Vous pouvez inclure autant de fichiers de sortie que vous le souhaitez (i.e., vous n'avez pas à fournir une solution pour chaque scénario d'entrée). Sur un système GNU/Linux, vous pouvez utiliser une ligne de commande comme celle-ci :

```
zip messolutions.zip life.*.out
```

Sur un système Windows, vous pouvez créer un fichier zip en sélectionnant

*Fichier* → *Nouveau* → *Dossier compressé (zippé)* au sein de l'explorateur Windows, puis copier vos fichiers de sortie dans ce nouveau fichier zip.

- Soumettez ce fichier zip sur le site du FARIO, de la même manière que vous soumettriez une solution classique.

Le site va regarder à l'intérieur du fichier zip et s'assurer que les fichiers de sortie sont correctement nommés, bien qu'il ne vérifiera pas le format ou la disposition de ces fichiers.

Si vous resoumettez un fichier zip différent, **l'ancien fichier zip sera supprimé**. Par exemple, supposons que vous soumettiez un fichier zip avec des solutions pour les neuf premiers scénarios, et que vous trouviez plus tard une solution pour le dixième. Votre solution doit contenir **l'ensemble des dix fichiers de sortie**, car lorsque vous soumettez vos nouvelles solutions, les anciennes seront supprimées.

Veuillez contacter les organisateurs [fario@fario.org](mailto:fario@fario.org) si vous avez des questions à propos de cette procédure.

## Problème 4

### Le Virus

**Limites de temps et de mémoire : 1 seconde, 32 MO**

Personne ne sait comment cela a commencé, c'est allé si vite. Il y a deux semaines, vous regardiez les informations au sujet de problèmes politiques de lointaines nations, et maintenant vous essayez d'échapper à vos propres voisins qui sont victimes du *Virus*. Il n'y a pas de thérapie connue pour le *Virus* qui se propage dans le pays – une fois que quelqu'un l'a contracté, il ne peut pas guérir. Heureusement, le Docteur Oui vous a contacté au sujet d'une curieuse arme biologique, qui est utilisée pour diffuser un vaccin contre le *Virus*. En utilisant cette arme, le vaccin peut être administré instantanément à tous les occupants d'une ville, ce qui a pour effet d'arrêter la propagation du *Virus* dans cette ville.

L'excentrique Docteur Oui vous explique frénétiquement son plan pour sauver le pays. Vous allez visiter toutes les villes du pays et utiliser l'arme dans chacune d'entre elles, en sauvant ceux que vous pouvez en vaccinant les personnes encore saines contre le *Virus*.

Les maires de chaque ville ont calculé la vitesse de diffusion du *Virus* dans leur ville, mesurée en victimes par heure. Vous devez choisir un plan de sauvetage de façon à minimiser le nombre de victimes. Certaines routes prennent plus de temps à parcourir que d'autres, et chaque heure gâchée signifie plus de personnes perdues à cause du *Virus*. Étant donnée une carte de toutes les routes du pays et le nombre d'habitants de chaque ville infectés par le *Virus* chaque heure, déterminez un itinéraire passant par toutes les villes, tel que le nombre total de personnes infectées par le *Virus* soit le plus faible possible. De plus, en ces temps d'insécurité avec des vols et meurtres fréquents, vous ne pouvez pas emprunter une route plus de deux fois.

Il y a  $N$  villes dans le pays, et  $N - 1$  routes à double sens qui relient les villes de façon à ce qu'il y ait exactement un chemin reliant chaque paire de villes. Vous savez, par vos études d'informatique, que le graphe ainsi formé est appelé un "arbre". Vous pouvez aussi supposer que les populations des villes sont suffisamment grandes pour que vous ne perdiez jamais tous leurs habitants à cause du *Virus*.

Vous vous apprêtez à commencer à coder, lorsque le Docteur Oui vous interrompt avec une dernière remarque.

"Souvenez-vous, des vies sont entre vos mains. Codez vite et vous serez le héros de demain."

#### Entrée

La première ligne de l'entrée contient un entier :  $N$ , le nombre de villes ( $1 \leq N \leq 100\,000$ ). Les villes sont numérotées de 1 à  $N$ . Vous devez commencer votre opération de sauvetage dans la ville 1.

La deuxième ligne de l'entrée contient  $N$  entiers positifs ou nuls. Le  $i$ ème entier représente le nombre d'habitants infectés par le *Virus* chaque heure dans la ville  $i$ . Ce nombre est au plus 100.

Suivent  $N - 1$  lignes d'entrée. Chaque ligne contient trois entiers séparés par des espaces :  $a$ ,  $b$  et  $h$ , représentant une route entre la ville  $a$  et la ville  $b$ , qui prend  $h$  heures à parcourir ( $1 \leq h \leq 100$ ).

Pour 60% des points disponibles,  $N \leq 1\,000$ .

Pour 20% des points disponibles,  $N \leq 10$ .

#### Sortie

La sortie doit contenir une ligne avec un entier : le nombre minimum de victimes du *Virus* avant que toutes les villes ne soient sauvées.

**Exemple d'entrée**

5  
 9 10 2 5 1  
 1 2 2  
 1 4 4  
 3 4 3  
 4 5 5

**Exemple de sortie**

101

**Explication**

L'itinéraire optimal est :

1  $\xrightarrow{2 \text{ heures}}$  2  $\xrightarrow{2 \text{ heures}}$  1  $\xrightarrow{4 \text{ heures}}$  4  $\xrightarrow{3 \text{ heures}}$  3  $\xrightarrow{3 \text{ heures}}$  4  $\xrightarrow{5 \text{ heures}}$  5

Ville	Heures avant sauvetage	Taux d'infection	Habitants infectés par le <i>Virus</i>
1	0	9	0
2	2	10	20
3	11	2	22
4	8	5	40
5	19	1	19
			Total : 101

**Calcul du score**

Le score pour chaque test d'entrée sera de 100 % si la bonne réponse est écrite sur la sortie, ou de 0 % sinon.