



Concours National d'Informatique

Sujet de demi-finale Nantes / Lyon

Samedi 18 mars 2006

CIMONE, ENCORE.

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre demi-finale. 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.
- 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 la demi-finale et la date, en haut de votre copie.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le nous, nous pouvons ajouter des questions plus difficiles.
- Tous les langages sont autorisés. Néanmoins, veuillez préciser celui que vous utilisez.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- 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.

2 Sujet

Introduction

Cimone (avec un “C”¹) est une adolescente extraordinaire. Vraiment. C’est le genre de fille perfectionniste, qui ne se laisse pas faire et qui réussit tout ce qu’elle entreprend. Elle a moult copines et son Q.I. est particulièrement élevé².

Cette histoire se passe un samedi après-midi. Par une quelconque coïncidence, ce jour se trouve être Mardi gras. Comme à son habitude, Cimone se défoule en jouant à son jeu préféré : Tétris[©]. Et encore une fois, elle fait un score plutôt impressionnant. Lassée par ce jeu qu’elle trouve désormais trop facile, elle se met à la recherche de nouveaux défis. Après de longues recherches sur Internet pour trouver d’autres jeux aussi intéressants, elle ne trouve que des jeux qu’elle maîtrise déjà parfaitement (Columns[©], Puzzle Bobble[©]...).

Cimone décide alors de créer son propre jeu. Elle s’inspire globalement de Tétris[©] : le concept de pièces qui tombent sur une grille était clairement une idée de génie. Il lui vient ensuite à l’esprit d’attribuer un numéro à chaque pièce, plutôt qu’une forme ou qu’une couleur. Après quelques heures de réflexion intense, elle a enfin défini les règles.

C’est ici que vous intervenez. En effet, malgré son caractère extraordinaire, Cimone a daigné faire appel à vous pour l’aider à développer son jeu.

L’interface du jeu sera donc une grille de NB_LIGNES x NB_COLONNES. Vous pouvez utiliser ces constantes dans vos algorithmes. Ensuite, au lieu de pièces de différentes formes comme à Tétris[©], ce sera des chiffres (de 0 à 9) qui tomberont.

Voici un exemple d’interface avec NB_COLONNES = 6 et NBLIGNES = 6.

```
012345
|-----|
|       |
|       |
|       |
|4      |
|221  6|
|311297|
|-----|
Next: 0
```

On note ici le champ “Next : 0” qui nous indique que le prochain chiffre qui tombera sera un ‘0’. Vous pourrez utiliser la constante variable PREVIOUS_NUMBER³ qui représente à tout moment le prochain chiffre qui tombera. Ici, nous avons PREVIOUS_NUMBER = 0.

A chaque tour, vous devez faire tomber un chiffre dans une colonne. Votre but est, comme à Tétris[©], de ne pas remplir la grille. Nous distinguerons un cas particulier : le cas du ‘0’. En effet, lorsqu’un ‘0’ tombe, il explose, faisant exploser les cases i et $i+1$ adjacentes (incluant les diagonales). De même, chaque case ayant explosée entraîne l’explosion des cases i et $i+1$ adjacentes. Une fois la réaction en chaîne des explosions terminées, la gravité reprend ses droits et fait tomber toutes les cases qui sont dans le vide. De cette façon, il n’y a jamais de trou.

¹Si.

²Par le plus grand des hasards, son Q.I. est très précisément égal au nombre de caractères de ce premier paragraphe.

³Si.

Reprenons l'exemple donné précédemment et faisons tomber le 0 dans la colonne '2' :

012345	012345	012345
-----	-----	-----
4 0	4 *	
221 6	*** 6	6
311297	****97	4 97
-----	-----	-----

Le '0' explose faisant exploser toutes les cases autour de lui (diagonales incluses).

Dans notre exemple, ça fait exploser la case qui contient un '1' qui est juste en dessous de lui et la case '2' qui est dans la diagonale inférieure gauche. Les explosions du '1' et du '2' font exploser toutes les cases autour (diagonales incluses) de valeur i et $i+1$ (avec $i = '1'$ ou $i = '2'$). Ces cases en explosant entraîneront à leur tour l'explosion des cases de valeur i et $i+1$ qui leur sont adjacentes (diagonale exclues⁴). Dans notre exemple, les cases qui exploseront sont marquées d'une '*'. Ensuite vient la gravité qui va faire tomber le '4' qui n'est plus posé sur rien.

Question 0 (0,5 point)

Quel est le Q.I. de Cimone ?

Question 1 (2 points)

Écrivez une structure de données permettant de représenter le jeu.

Cette question est importante car vous devrez utiliser votre structure pour répondre aux questions suivantes. Il est donc conseillé de lire l'intégralité du sujet avant d'y répondre.

Question 2 (2 points)

Écrivez une fonction qui calcule le nombre de cases de la grille.

Question 3 (3,5 points)

Écrivez une fonction qui renvoie la hauteur de la pièce la plus haute du jeu.

Question 4 (4 points)

Partie 1

Écrivez une fonction qui applique la gravité sur la zone de jeu (on considère qu'il vient d'y avoir une explosion).

Partie 2

Écrivez une fonction qui met à jour la grille après une explosion. Votre fonction prends les coordonnées de l'explosion en paramètres.

⁴Mais non, c'est une blague

Question 5

Partie 1 (5 points)

Écrivez une fonction qui joue du mieux possible.

Partie 2 (2 points)

Etant donné que la principale plateforme de jeu de Cimone est son téléphone portable, il faudra que votre programme puisse s'exécuter de manière fluide sur celui-ci. Cependant, Cimone ne possède pas n'importe quel téléphone portable. Il y a quelques années de cela, un fabricant de téléphones portables a remarqué Cimone et le directeur a eu une idée qui allait révolutionner le marché. Le téléphone est cadencé à la vitesse du QI de l'utilisateur ! Le téléphone de Cimone est donc cadencé à `QI_DE_SIMONE` Mhz. D'autre part, les technologies évoluant, la mémoire du téléphone utilise directement la mémoire de Cimone. Etant donné que celle-ci a une mémoire infinie, vous n'avez pas de limite de mémoire.

Combien de temps mettra votre programme à s'exécuter sur le téléphone de Cimone ?

Combien de temps mettra votre programme à s'exécuter sur le même modèle de téléphone mais avec vous au lieu de Cimone comme utilisateur ?

Question 6 (points bonus)

Ces questions peuvent vous rapporter des points seulement si vous avez répondu juste à toutes les questions précédentes.

Voici un plateau de jeu, faites jouer votre algorithme de la question 5 et ensuite tentez de faire mieux à la main. Si vous y arrivez, quels points pensez vous devoir modifier à votre algorithme pour qu'il soit plus performant ? On considère comme plus performant l'algorithme qui obtient en résultat le plateau de jeu contenant le moins de chiffres possible.

```
0123456789
|-----|
|4       |
|3       |
|6512    |
|4398   894|
|9854423432|
|5498189431|
|-----|
```

Liste des chiffres à tomber dans l'ordre : 3 2 9 4 8 3 0 9 5 0 8 1 0

(+1 point présentation)