



Concours National d'Informatique

**Algorithmique**  
**Demi-finale Lyon / Toulon**

Samedi 15 février 2003

# ALCATRAZ



## 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, 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 la demi-finale 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.
- 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.
- Écoutez bien ce que disent les organisateurs, mais ne les croyez pas !

## 2 Sujet

### Introduction

Vous êtes emprisonné depuis des années dans une prison de haute sécurité, et venez enfin de trouver comment sortir de votre cellule, et passer les différentes portes sécurisées qui vous séparent du monde extérieur. La seule chose qui vous pose encore problème, ce sont les gardes !

Depuis le temps que vous vivez là, vous connaissez parfaitement le comportement de tous les gardes, et pouvez donc prédire leurs déplacements. Il ne reste plus qu'à trouver comment utiliser ces données pour vous échapper sans qu'aucun d'entre eux ne vous voie.

La prison peut être représentée par une grille rectangulaire, contenant deux types de cases : des cases pleines, représentant des murs infranchissables, et des cases vides, représentant des zones par lesquelles vous pouvez passer, d'une manière ou d'une autre. La sortie de la prison est également fournie, par ses coordonnées.

Pour représenter vos déplacements, ainsi que le déplacement des gardes, on décompose le temps en tours. Vous comme les gardes, pouvez vous déplacer d'une case par tour, horizontalement ou verticalement sur la grille, mais pas en diagonale.

Chaque garde se déplace toujours de la même manière. Vous connaissez :

- Sa position de départ (coordonnées  $x$  et  $y$ ).
- La succession de ses déplacements, pendant tout son cycle de garde.

Changer de direction ne demande pas de temps à un garde : à la fin d'un déplacement, il se tourne automatiquement dans la direction du déplacement suivant.

A la fin de cette succession de déplacements, il se retrouve à sa position de départ, et recommence. A chaque tour, un garde se déplace dans une des 4 directions, ou bien reste sur place.

Chaque garde peut voir autour de lui dans 3 directions, et sur une distance de 3 cases. Il regarde dans la direction de son prochain déplacement, et peut voir devant lui, à gauche, ou à droite.

Sur le schéma suivant, G représente la position du garde, et O les cases qu'il peut voir à partir de cette position, sachant qu'il va se déplacer vers le haut dans 2 tours (il reste sur place au tour prochain).

```
XXXXXXXXXX
XXXOXOXXX
XXXOXOXXX
XXXOXOXXX
XOOOGOOX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

Les murs ne sont pas transparents, donc le champ de vision du garde est bloqué dès qu'il passe par un mur.

On définira les constantes `MAX_JAIL_SIZE` (valant 200), `MAX_GUARDS` (valant 20), et `MAX_MOVES` (valant 1000), qui représentent respectivement le nombre maximal de lignes/colonnes de la prison, le nombre maximum de gardes, et le nombre maximum de déplacements d'un garde avant qu'il ne retrouve sa position de départ et ne recommence.

### Question 1 (2 points)

Définissez les structures de données permettant de stocker :

- votre position dans la prison ;
- la structure de la prison, avec ses murs, et sa sortie ;
- le comportement des gardes : leur position de départ, et leurs déplacements.

### Question 2 (3 points)

On vous donne le comportement d'un garde, et la structure de la prison, ainsi qu'une date (en nombre de tours depuis le départ).

Écrivez une fonction qui détermine où se trouve ce garde à la date donnée.

### Question 3 (3 points)

On vous donne la position d'un garde, la direction de son prochain déplacement, la structure de la prison, et votre position.

Écrivez une fonction qui détermine si le garde peut vous voir là où vous êtes.

### Question 4 (4 points)

On vous donne la structure de la prison, le comportement des gardes, et votre déplacement, au même format que le comportement d'un garde, à part qu'il ne boucle pas : vous vous arrêtez simplement à la fin du déplacement.

Écrivez une fonction qui détermine si ce déplacement vous amène sans danger à la sortie. (sans être vu par les gardes)

### Question 5

#### • Partie 1 (5 points)

On vous donne la structure de la prison, votre position, et le comportement des gardes.

Écrivez une fonction qui détermine s'il existe un chemin qui vous permet d'atteindre la sortie, sans être vu par les gardes. Votre fonction doit simplement indiquer si ce chemin existe.

#### • Partie 2 (2 points)

Vous disposez d'une machine disposant de 16Mo de mémoire, et d'un processeur à 100 mhz. Déterminez grossièrement le temps que prendra votre fonction à s'exécuter, sachant que les dimensions de la prison sont de 200 par 200, et qu'elle contient 20 gardes. Justifiez votre réponse.

Si vous avez répondu à toutes les questions, relisez bien. Si vous êtes sûr que tout est bon, et qu'il vous reste encore du temps, demandez à un organisateur de vous donner une question supplémentaire.

**Question 6** (2 points)

On vous donne la structure de la prison, votre position, et le comportement des gardes.

Vous avez trouvé un moyen de vous débarasser temporairement de 4 des gardes (au choix).

Ecrivez une fonction qui détermine s'il existe un chemin qui vous permet d'atteindre la sortie, après vous être débarassé de 4 gardes. La fonction doit simplement indiquer si c'est possible ou non.

**Question 7** (1 points)

Même question que la partie II de la question 5, dans les mêmes conditions.

(+1 point présentation)