



Concours National d'Informatique  
**Algorithmique**  
Demi-finale Paris / Strasbourg

Samedi 25 janvier 2003

# FILES D'ATTENTES À DISNEYLAND



## 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 vous apprêtez à passer la journée à Disneyland, mais vous avez mal choisi votre jour : c'est plein à craquer, et il y a de longues files d'attente devant toutes les attractions !

Heureusement, le parc est moderne, et propose un tableau d'affichage, où pour chaque attraction, et chaque moment de la journée, vous pouvez lire combien de temps il faudra probablement attendre si vous entrez dans la file d'attente à ce moment. Ce tableau est décomposé en plages de 10 minutes, pour toute la journée (de 9h à 21h). Lorsqu'il est trop tard pour voir une attraction avant que le parc ne ferme, le tableau affiche une étoile à la position correspondante. Dans les autres cas, le temps est un nombre entier de minutes, pouvant aller jusqu'à 2h, soit 120 minutes.

Vous souhaitez profiter de ces informations au maximum, pour rentabiliser votre journée.

Les différentes attractions vous intéressent plus ou moins. Vous attribuez une note entre 0 et 20 à chacune d'entre-elles, en fonction de l'intérêt que vous lui trouvez. Pour chaque attraction, vous connaissez également sa durée : le temps que vous passerez dans l'attraction elle-même, qui est un nombre entier de minutes. (20 minutes maximum)

Dans tout le problème, on négligera le temps nécessaire pour aller d'une attraction à l'autre.

Certaines attractions n'ont d'intérêt que la première fois que vous les faites, et il est hors de question que vous les fassiez deux fois dans la journée.

### Question 1 (2 points)

Définissez les structures de données permettant représenter les informations suivantes :

- les différentes attractions, et pour chacune, sa durée, l'intérêt que vous lui portez, et le fait qu'elle puisse ou non être faite plusieurs fois.
- le tableau d'affichage, représentant les temps d'attente aux différents moments de la journée.
- le parcours d'une personne : la succession des attractions qu'elle souhaite voir dans sa journée, dans l'ordre.

On définira la constante `NB_ATTRACTION`, valant 200 : le nombre d'attractions maximum à gérer pour le parc.

### Question 2 (3 points)

On vous fournit le contenu du tableau d'affichage.

Ecrivez une fonction qui détermine le meilleur moment pour voir une attraction donnée, c'est-à-dire le moment où l'attente sera la moins longue.

### Question 3 (3 points)

On vous donne le contenu du tableau d'affichage.

Vous êtes un fan d'une attraction, et souhaitez la voir 10 fois de suite. C'est-à dire que dès que vous l'avez faite, vous retournez immédiatement dans la file d'attente pour recommencer. Ecrivez une fonction qui détermine à quel moment il vaut mieux commencer cette série, pour que l'ensemble prenne le moins de temps possible.

### Question 4 (4 points)

On vous donne votre parcours de la journée, le contenu du tableau d'affichage, et la description de toutes les attractions.

Ecrivez une fonction qui détermine le temps qu'il faut pour effectuer ce parcours. Cette fonction doit renvoyer -1 si ce parcours n'est pas valide, c'est-à dire qu'il vous fait passer plusieurs fois par une attraction que vous ne souhaitez voir qu'une fois.

### Question 5

#### • Partie 1 (5 points)

On vous donne la descriptions des attractions, ainsi que le contenu du tableau d'affichage.

Ecrivez une fonction qui détermine le meilleur parcours possible dans cette journée, c'est-à dire celui qui permet d'accumuler un total de points d'intérêt des attractions le plus grand possible. Toutes les attractions peuvent être vues plusieurs fois (et leur nombre de points compté autant de fois)

#### • Partie 2 (2 points)

Vous disposez d'un palm ayant d'un processeur à 100 Mhz, et de 16Mo de mémoire. Déterminer un ordre de grandeur du temps qu'il faudra à votre fonction pour s'exécuter sur cette machine, sachant qu'il y a 200 attractions dans le parc. On vous rappelle que la journée va de 9h à 21h, donc contient 720 minutes.

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 description des attractions, ainsi que le tableau d'affichage.

Parmi les attractions, 6 sont à voir une fois et une seule. Ecrivez une fonction qui détermine le total de points que vous pouvez obtenir au maximum dans la journée, en passant par des attractions de cette liste.

Les conditions sont identiques à celles de la partie II de la question 5. Estimez le temps que prendra votre nouvelle fonction pour s'exécuter.

**Question 7** (1 point)

Les conditions sont identiques à celles de la partie II de la question 5. Estimez le temps que prendra votre nouvelle fonction pour s'exécuter.

(+1 point présentation)