

Le démon de Tetris

Si vous avez déjà joué à Tetris, n'avez-vous jamais rêvé de voir dans le futur quelle séquence de pièces vous réserve ce logiciel démoniaque? "Ah si seulement j'avais su qu'il ne me donnerait pas de barre pendant si longtemps, je n'aurais pas joué de cette façon!", avez-vous certainement hurlé maintes fois!

Votre rêve est maintenant exaucé ! Aujourd'hui, vous (ou plutôt votre programme) serez le démon de Tetris : tel le démon de Laplace, vous aurez connaissance du passé, du présent et du futur pour prendre vos décisions. Serez-vous alors capables d'atteindre les plus hauts scores jamais réalisés par un joueur de Tetris?

Le sujet est découpé en 5 questions. Les questions sont toujours de la même forme: on vous fournit des fichiers d'entrées téléchargeables et vous devez soumettre en échange des fichiers de sorties au serveur de correction. Un programme sur le serveur évaluera alors le nombre de points acquis en lisant vos fichiers de sortie. Attention à bien respecter les noms de fichier et les formats attendus!

Par ailleurs, ne vous échangez pas les fichiers de sortie car à la fin de la compétition, le jury se réunira pour lire votre code et vous devrez nous permettre de reconstruire les fichiers de sorties à partir des fichiers d'entrées en lançant la commande:

```
bash ./play.sh
```

depuis la racine de votre archive. Dans tous les cas, notez bien que le correcteur automatique ne considérera que les fichiers "output" fournis dans l'archive.

QUESTION 1 : LA DESTRUCTION DU TEMPS

Commençons par nous entendre une fois pour toute sur la fameuse séquence de pièces. À partir d'un unique nombre entier s (compris en 0 et 1000000), nous allons nous donner un moyen de calculer une séquence τ de nombres entiers compris entre 1 et 7 (un nombre par pièces de Tetris).

Cette séquence est obtenue comme suit :

1. On calcule la séquence u dont le premier terme est s et dont le terme de rang $N+1$ est obtenu à l'aide du terme A de rang N par la formule $(22695477 * A + 1) \bmod (2^{32})$. Notez que l'on obtient alors un nombre dont seuls 32 bits sont significatifs.
2. q est la séquence de nombres entiers compris entre 0 et 7 dont le terme de rang N est obtenu à l'aide des 3 bits de poids les plus forts du terme de rang N de u .
3. τ est obtenue en ne gardant que les termes non nuls de q .

Pour chaque fichier d'entrée dont le nom est de la forme `q1-input-k` (avec k un entier), on vous demande de fournir un fichier nommé `q1-output-k`. Le fichier `q1-input-k` est formé d'une unique ligne contenant l'entier s . Le fichier `q1-output-k` doit répéter l'entier s sur sa première ligne puis contenir la suite des 20 premiers termes de la suite τ correspondant à cet entier s , un par ligne.

Ainsi, si `q1-input-73` contient 231, le fichier `q1-output-73` doit contenir:

```
231
1
5
6
2
6
5
6
2
5
4
1
3
3
7
6
5
7
3
1
5
```

QUESTION 2 : DES BLOCS À LEUR PLACE

Un plateau de jeu Tetris est une matrice de blocs composée de 24 lignes et de 12 colonnes. Les lignes sont numérotées du haut vers le bas de 0 à 23. Les colonnes sont numérotées de gauche à droite de 0 à 11.

Un bloc est représenté par un entier de 0 à 7. 0 est

le bloc vide tandis que les blocs entre 1 et 7 correspondent à des blocs inclus dans des pièces de Tetris du même numéro.

En effet, les pièces sont numérotées comme l'indique le schéma suivant:

Pièce 1	1111
Pièce 2	222 2
Pièce 3	333 3
Pièce 4	44 44
Pièce 5	55 55
Pièce 6	666 6
Pièce 7	77 77

On dit que l'on "tourne une fois" une pièce si on lui fait faire un quart de tour dans le sens horaire. Ainsi, voici comment tourne la pièce 2 :

Une fois :	2 2 22
Deux fois :	2 222
Trois fois :	22 2 2
Quatre fois :	222 2

Pour placer une pièce tournée R fois sur le plateau à la ligne L et à la colonne C , il faut l'avoir préalablement tournée R fois puis placer ses blocs de façon à ce que son bloc le plus haut se trouve sur la ligne L et que son bloc le plus à gauche se trouve sur la colonne C .

Une pièce peut être posée sur le plateau seulement si tous ses blocs le peuvent. Un bloc peut être posé sur le plateau seulement si l'emplacement du plateau est un bloc vide et si le bloc n'est pas situé à l'extérieur du plateau.

Dans cette question, chaque fichier d'entrée est nommé `q2-input-k` (avec k un entier) et est formé de lignes contenant 4 entiers naturels séparés par un espace. Le premier entier est le numéro P de la pièce (compris entre 1 et 7). Le second entier est le numéro L de la ligne (compris entre 0 et 23). Le troisième entier est le numéro C de la colonne (compris entre 0 et 11). Le quatrième entier est le nombre de tours R appliqués à la pièce (compris entre 0 et 3). On part du plateau ne contenant que des blocs vides et on interprète chacune de ses lignes comme suit :

- si on peut poser la pièce P en ligne L et colonne C après l'avoir fait tourner R fois alors on la place sur le plateau ;
- sinon, on note que cette ligne du fichier d'entrée est incorrecte.

Le fichier de sortie, nommé `q2-output-k`, doit commencer par répéter les lignes qui sont incorrectes, dans le même ordre que leur apparition dans le fichier d'entrée, puis il doit décrire le plateau obtenu après le traitement de toutes les lignes.

Pour décrire le plateau, on affichera les lignes par ordre croissant de numéro. Les blocs de chaque ligne sont séparés par un espace. On affichera le caractère `.` pour représenter un bloc vide et l'entier de son code pour un bloc non vide.

Par exemple, pour `q2-input-37` valant:

```
1 5 5 0
1 5 5 0
1 6 5 0
1 6 8 0
2 7 0 0
2 7 4 0
2 8 0 0
2 9 0 0
3 11 0 0
4 12 0 0
```

On obtient le fichier `q2-output-37` suivant:

```
1 5 5 0
1 6 8 0
2 8 0 0
.....
.....
.....
.....
.....1111...
.....1111...
222.222.....
..2...2.....
222.....
..2.....
```

```
333.....  
344.....  
44.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
```

QUESTION 3 : LA GRAVITÉ EST UNE LOI UNIVERSELLE

On suppose dans cette question que les quatre premières lignes du plateau sont toujours vides. On dit qu’une pièce tournée R fois est lâchée sur le plateau à la colonne c si on la pose sur la première ligne et qu’on la déplace d’une ligne vers le bas tant que c’est possible. (Ici, un déplacement est “possible” si on peut poser la pièce au sens de la question précédente.)

Dans cette question, chaque fichier d’entrée est nommé `q3-input-k` (avec k un entier). Sur chaque ligne, on trouve trois entiers séparés par un espace. Le premier entier P est un numéro de pièce. Le second entier C est un numéro de colonne. Le troisième entier R est le nombre de fois où la pièce a été tournée. On peut supposer que l’on peut toujours poser la pièce P sur la première ligne et la colonne C en l’ayant fait tourner préalablement R fois. (Il existe des couples (C, R) invalides mais nous ne les utilisons pas dans cette question. En d’autres termes, on suppose dans cette question que l’on peut toujours placer les pièces sur les quatre premières lignes avant de les faire descendre.)

Le fichier de sortie doit être de la forme `q3-output-k`. Il doit décrire le plateau obtenu après interprétation de toutes les lignes du fichier d’entrée dans l’ordre de leur apparition dans le fichier. On utilisera pour cela le même format que celui de la question 2.

Ainsi, si le fichier d’entrée est `q3-input-73` de contenu:

```
1 0 0  
2 1 0  
3 2 0  
4 3 0  
5 4 0  
6 5 0  
7 6 0  
1 7 0
```

```
2 8 0
3 9 0
```

Le fichier de sortie obtenu doit être q3-output-73 et contenir:

```
.....
.....
.....
.....
.....
.....
.....
.....333
.....3..
.....222
.....2..
.....1111
.....77...
.....77...
.....666...
.....556...
.....55...
.....44...
.....44...
.....333...
.....3...
.....222...
.....2...
1111.....
```

QUESTION 4: CALCUL DU SCORE

Nous avons maintenant tous les ingrédients pour jouer à Tetris.

Supposons donné un nombre s : la séquence τ (de la question 1) déterminée par ce nombre s nous donne la séquence de pièces présentées au joueur.

Le joueur doit décider sur quelle colonne et combien de fois tourner chaque pièce de cette séquence. Avant de jouer chaque pièce, on vérifie que les quatre premières lignes sont bien vides : dans le cas contraire, la partie est terminée.

La pièce est alors lâchée où l'a décidé le joueur. Si on ne peut pas lâcher la pièce, alors la partie est terminée.

Ensuite, il se passe la magie de Tetris : si une ligne du plateau est remplie de blocs non vides alors elle est éliminée du plateau et toutes les pièces au-dessus descendent d'une ligne. Si quatre lignes contiguës sont éliminées, le joueur gagne 800 points : c'est un Tetris! Si la pièce précédente avait déjà donné lieu à un Tetris, alors il gagne même 1200 points! Sinon, chaque ligne remporte 100.

Dans cette question, le fichier d'entrée est de la

forme `q4-input-k` (avec k un entier naturel). La première ligne de ce fichier contient le nombre s . Ensuite, la ligne numéro $N+1$ est formée de 3 entiers.

Le premier entier p est le numéro de la pièce : il doit correspondre au terme de rang N de la suite τ (de la question 1) initialisée par S . Sinon, la partie est terminée.

Le second entier c est la colonne où le joueur a décidé de lâcher sa pièce. Le troisième entier r est le nombre de fois où la pièce est tournée.

Si on ne peut pas lâcher la pièce sur le plateau (parce que le couple (c, r) fourni la fait sortir du plateau) alors la partie est terminée. Sinon, on lâche la pièce sur le plateau et on met à jour le score comme indiqué plus haut.

On passe à la ligne suivante. S'il n'y a plus de ligne, alors la partie est terminée.

On limite le nombre de lignes à 10000.

Le fichier de sortie doit être de la forme `q4-output-k` et doit d'abord décrire le plateau obtenu lorsque la partie est terminée après avoir évalué les lignes du fichier `q4-input-k` dans leur ordre d'apparition dans ce fichier. Cette description doit respecter de nouveau le format décrit dans la question 2. La dernière ligne de ce fichier doit contenir le score obtenu par le joueur.

Ainsi, si `q4-input-73` contient

```
31
1 8 2
3 7 0
5 4 2
3 1 2
4 1 2
4 10 3
7 8 3
7 6 3
5 3 2
2 0 3
1 8 2
6 4 0
1 0 2
3 7 0
5 0 2
7 10 3
4 5 2
6 3 0
3 9 0
6 1 1
6 7 1
1 3 2
1 4 2
5 0 3
6 9 2
```

```
4 2 3
5 7 2
2 0 3
3 10 1
3 5 1
4 3 3
6 8 0
6 0 2
7 6 3
3 4 1
3 0 0
2 9 0
4 9 2
6 2 0
6 7 1
7 5 3
4 0 2
7 3 3
3 9 0
3 0 0
1 5 2
1 8 2
6 6 1
4 4 2
4 2 2
2 9 2
1 5 2
2 1 3
7 10 3
3 4 0
7 8 3
6 2 1
6 0 3
4 6 3
2 9 0
5 3 2
6 1 1
6 7 0
2 3 0
2 9 0
7 0 3
4 7 2
6 1 0
2 4 0
3 0 0
7 10 3
3 7 0
6 4 2
1 1 2
4 9 2
3 6 0
1 2 2
6 0 3
2 7 2
7 5 3
5 2 2
5 9 2
7 0 3
3 4 0
7 7 3
1 3 2
4 10 3
```

alors le fichier q4-output-73 doit contenir:

```
.....
.....
.....
.....
```



```
.....  
.....  
.....  
.....4.  
...111177.44  
77..33377554  
77553772..55  
6..557722244  
66111133344.  
333.66634477  
776222666.2  
7765524.6222  
6666554477.2  
.2644446.222  
.24444661111  
44666776644.  
36.4.3776662  
224.433.5563  
2443553333.4  
2333.5531111  
1500
```

QUESTION 5: À VOUS DE JOUER!

Voilà, votre moment de gloire arrivé : si on vous donne l'entier s , vous connaissez la séquence des pièces présentées par Tetris. C'est le moment d'appliquer votre stratégie démoniaque pour obtenir le meilleur score!

Dans cette question, le fichier d'entrée est de la forme `q5-input-k` et contient seulement l'entier s .

Le nom du fichier de sortie est de la forme `q5-output-k` et doit suivre le même format que les fichiers d'entrées de la question 4.

Le score obtenu en évaluant le contenu de `q5-output-k` déterminera les points de cette question et votre classement final!