
Olympiades Régionales
d'Informatique Franco-Australiennes
14 mars 2010

Durée : 4 heures

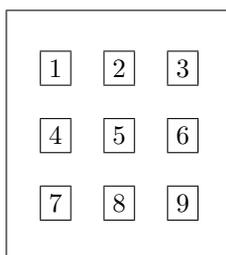
4 problèmes

Problème 1

Mission FARIO

Limites de temps et de mémoire : 1 seconde, 16 Mo

Vous êtes un super-espion. Ce métier vous oblige à voyager partout dans le monde, à visiter des endroits exotiques et à conduire des voitures de luxe. Votre dernière mission est de pénétrer dans le siège de l'organisation malfaisante F.A.R.I.O. et d'y dérober des documents importants. Entrer est une tâche facile pour un agent comme vous, mais les documents sont dans un coffre-fort qui ne peut être ouvert qu'en tapant la bonne combinaison.



Le clavier du coffre-fort est une grille 3×3 de touches numérotées de 1 à 9, comme ci-dessus. Vos sources vous ont dit que *deux chiffres consécutifs dans le code seront toujours des touches adjacentes du clavier*. Par exemple, le chiffre 1 ne sera suivi que par un 2 ou un 4, le chiffre 2 ne sera suivi que par un 1, un 3 ou un 5, etc. Ainsi 3252 et 12369 sont des codes valides, mais 1234 ne l'est pas (3 n'est pas adjacent à 4 sur le clavier) et 55 non plus (5 n'est pas adjacent à 5 sur le clavier).

Vous connaissez également le nombre de chiffres du code, ainsi que le premier chiffre. Étant données ces informations, combien y a-t-il de codes secrets possibles ?

Entrée

Votre programme doit lire sur l'entrée standard. L'entrée est composée de deux entiers séparés par un espace, N et D . N est le nombre de chiffres du code ($1 \leq N \leq 1\,000\,000$), et D est le premier chiffre ($1 \leq D \leq 9$).

Pour 30% des points disponibles, $N \leq 10$.

Sortie

Votre programme doit écrire une seule ligne sur la sortie standard, comportant un unique entier : le nombre de codes possibles, *modulo* 1 000 003.

Exemple d'entrée 1

3 4

Exemple d'entrée 2

42 8

Exemple de sortie 1

8

Exemple de sortie 2

128254

Commentaires

Dans l'exemple 1, le code est de longueur 3 et commence par le chiffre 4. Les codes possibles sont 412, 414, 452, 454, 456, 458, 474 et 478.

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.

Problème 2

ABC...

Limites de temps et de mémoire : 5 secondes, 32 Mo

Malgré tous vos efforts, vous n'arrêtez pas de faire des fautes de frappe. Vous essayez sans cesse de vous améliorer mais vous trouvez cela vraiment difficile. Heureusement vous avez trouvé une solution géniale : vous allez écrire un programme pour corriger vos fautes !

Votre « vérificateur de fautes » doit être capable de prendre un dictionnaire de mots valides et un message, et de corriger toutes les fautes de frappe qui s'y trouvent. Dans ce problème, nous considérerons qu'une *faute de frappe* est un mot qui n'est pas dans le dictionnaire, et à partir duquel on peut obtenir un mot valide en remplaçant une et une seule lettre par une lettre différente.

Entrée

Votre programme doit lire sur l'entrée standard. La première ligne de l'entrée contient un entier N , le nombre de mots du dictionnaire ($1 \leq N \leq 100\,000$). Les N lignes suivantes contiennent chacune un mot du dictionnaire. Chaque mot du dictionnaire n'apparaît qu'une fois.

La ligne suivante contient un entier M , le nombre de mots du message ($1 \leq M \leq 10\,000$). Les M lignes suivantes contiennent chacune un mot devant être vérifié. Tous les mots seront composés de 1 à 20 lettres minuscules non accentuées.

Pour 30% des points disponibles, $N, M \leq 1000$. Pour 60% des points disponibles, $N \leq 30\,000$.

Sortie

Votre programme doit écrire M lignes sur la sortie standard, une pour chacun des mots du message :

- Si le mot en entrée est correctement orthographié (c'est-à-dire qu'il est dans le dictionnaire), écrivez-le tel quel ;
- Si le mot en entrée peut être corrigé en remplaçant un seul caractère, écrivez la version corrigée. S'il y a plusieurs possibilités, écrivez n'importe laquelle d'entre elles ;
- Si le mot en entrée ne peut pas être corrigé en remplaçant un caractère, écrivez uniquement le caractère « ? ».

Exemple d'entrée

```
9
i
apple
far
mat
job
for
then
may
apply
6
may
i
apple
fur
the
jpb
```

Exemple de sortie

```
may
i
apple
for
?
job
```

Calcul du score

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

Problème 3

Site de rencontres en ligne

Limites de temps et de mémoire : 3 secondes, 64 Mo

Vous dirigez un site de rencontres en ligne avec de nombreux utilisateurs. Vous gardez des statistiques détaillées de chaque utilisateur, entre autres le nombre de fois qu'il a visité votre site dans le mois. Vous aimeriez savoir combien de fois votre site a été visité au total ce mois-ci. Malheureusement, ce n'est pas aussi facile que ça en a l'air – certains de vos utilisateurs sont des *spammeurs* qui utilisent votre site pour faire de la publicité pour leurs produits. Vous ne voulez pas compter les visites des spammeurs dans votre total.

Lorsqu'un utilisateur s'inscrit sur le site, il doit fournir un certain nombre d'informations personnelles dont son âge, sa taille et son poids. Les spammeurs mentent à propos de ces informations, mais ils sont très prévisibles et vous pouvez utiliser leurs réponses fausses pour les repérer. Par exemple, vous savez peut-être que « tout utilisateur dont l'âge est entre 18 et 30, dont la taille est entre 80 cm et 90 cm, et dont le poids est entre 40 kg et 50 kg est un spammeur ».

Votre objectif est d'écrire un programme qui prend un certain nombre de déclarations comme celle ci-dessus et une liste des statistiques des utilisateurs, et qui calcule le nombre total de visites de votre site au cours du mois.

Entrée

Votre programme doit lire sur l'entrée standard. Dans ce qui suit, a_i , h_i , w_i , v_i représentent pour un utilisateur, respectivement son âge, sa taille, son poids et le nombre de ses visites dans le mois.

La première ligne de l'entrée contient deux entiers séparés par un espace, S et U ($1 \leq S, U \leq 1\,000\,000$). Chacune des S lignes suivantes est de la forme « $a_{min} a_{max} h_{min} h_{max} w_{min} w_{max}$ », représentant la déclaration « tout utilisateur qui satisfait à la fois $a_{min} \leq a_X \leq a_{max}$ et $h_{min} \leq h_X \leq h_{max}$ et $w_{min} \leq w_X \leq w_{max}$ est un spammeur » (bien sûr, $a_{min} \leq a_{max}$, etc). Chacune des U lignes suivantes est la description d'un utilisateur, sous la forme « $a_i h_i w_i v_i$ ».

Âges, tailles et poids seront tous des entiers compris entre 0 et 99 inclus. Chaque utilisateur fait entre 1 et 1000 visites.

Référez-vous à la section *Calcul du score* pour voir la répartition des points attribués pour les différents fichiers tests.

Sortie

Votre programme doit écrire une seule ligne sur la sortie standard, comportant un unique entier : le nombre de visites de votre site ce mois-ci par des utilisateurs qui ne sont *pas* des spammeurs.

Exemple d'entrée

```
2 3
18 24 30 60 80 85
21 30 40 50 0 99
18 30 80 3
25 35 86 5
23 45 82 7
```

Exemple de sortie

5

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.

Le tableau ci-dessous répertorie la taille des entrées utilisées dans les différents fichiers tests. Ici, $N - 1$ représente la valeur maximale de l'âge, la taille et du poids au sein du fichier test.

N	S	U	Pourcentage des points
20	10	1 000	10 %
20	100	8 000	10 %
10	10 000	100 000	10 %
30	10 000	1 000 000	10 %
50	100 000	100 000	20 %
100	1 000 000	500 000	20 %
100	1 000 000	1 000 000	20 %

Problème 4

Écran LCD humain

Limites de temps et de mémoire : 10 secondes, 64 Mo

Vous faites tout votre possible pour rendre les concours d'informatique plus distrayants pour le public. Cette année, vous organisez les olympiades dans un gigantesque stade avec de nombreux spectateurs. Alors que les candidats tenteront de se concentrer sur les problèmes, des pom-pom girls pousseront des cris d'encouragement tandis que les scores apparaîtront en temps réel sur des écrans géants. La mascotte des olympiades lancera de petits cadeaux à la foule et les guides formeront un immense écran LCD humain.

Tout cela représente un coût plus élevé que prévu et vous avez dû trouver un sponsor supplémentaire à la dernière minute. Une chaîne de restauration rapide a accepté de vous aider, à la condition que leur logo soit la dernière image du clip LCD humain.

À la fin de la version initialement prévue du clip, les guides formeront une superbe grille 2D. Chaque guide, portant un T-shirt coloré, constituera une cellule de la grille, formant ainsi un pixel de l'écran LCD humain. On vous donne la description de cette grille et la couleur du guide de chaque cellule. On vous donne également la couleur que chaque cellule devrait afficher pour représenter parfaitement le logo du sponsor.

Comme vous n'avez pas beaucoup de temps pour entraîner les guides à ce nouveau mouvement, vous voulez faire les choses simplement : chaque guide devra soit rester immobile, soit échanger sa position avec l'un des guides des 4 cellules adjacentes. Tous les échanges se produiront simultanément et aucun guide ne devra effectuer plus d'un échange (contrairement aux vrais écrans LCD, les guides ne peuvent pas changer de couleur).

Votre but est de trouver un ensemble d'échanges tels que l'image produite sur la grille est aussi proche que possible du logo du sponsor.

Entrée

La première ligne de l'entrée contient deux entiers séparés par un espace R, C : le nombre de lignes et le nombre de colonnes de la grille ($1 \leq R, C \leq 100$).

Les R lignes suivantes décrivent la grille telle qu'elle apparaît au public à la fin du clip initial. Chaque ligne décrit une ligne de la grille et contient C entiers séparés par des espaces, correspondant aux couleurs des guides formant les cellules de cette ligne. Toutes les couleurs sont des entiers compris entre 0 et 9 inclus.

Les R lignes suivantes décrivent la grille que vous essayez d'obtenir, qui représente le logo du sponsor. Chaque ligne décrit une ligne et contient C entiers séparés par des espaces, correspondant aux couleurs que les cellules doivent afficher pour former le logo du sponsor.

Référez-vous à la section *Calcul du score* pour voir la répartition des points attribués pour les différents fichiers tests.

Sortie

Votre programme doit écrire plusieurs lignes sur la sortie standard. La première ligne doit contenir un unique entier S , le nombre d'échanges de votre solution.

Chacune des S lignes suivantes doit contenir quatre entiers : les coordonnées (r_1, c_1) et (r_2, c_2) de deux guides adjacents qui devront échanger leurs positions. La cellule du coin en haut à gauche a pour coordonnées $(0, 0)$.

Exemple d'entrée

```
3 4
1 2 3 4
4 3 2 1
1 1 2 2
4 3 2 4
1 1 1 2
2 3 2 2
```

Exemple de sortie

```
4
0 0 1 0
0 1 0 2
1 1 2 1
1 2 1 3
```

La grille suivante est produite en réalisant les échanges donnés dans l'exemple :

```
4 3 2 4
1 1 1 2
1 3 2 2
```

Sur les 12 cellules, 11 correspondent au logo du sponsor. C'est une solution optimale.

Calcul du score

Le score de chaque fichier test sera calculé sur la base du nombre de cellules qui correspondent au logo, comme ci-dessous :

- Si la séquence d'échanges est invalide (par exemple si un mouvement invalide est effectué ou si un guide fait plusieurs échanges) alors le score pour ce test sera de 0 %.
- Si le nombre de cellules qui correspondent au logo est optimal (c'est-à-dire qu'aucun autre ensemble d'échanges ne donne un résultat avec plus de cellules correspondant au logo), alors le score pour ce test sera de 100 %.
- Si le nombre de cellules qui correspondent au logo est inférieur ou égal à celui de l'image initiale alors le score pour ce test sera de 0 %.
- Dans les autres cas, soit a le nombre de cellules correspondant au logo dans une solution optimale. Soit b le nombre de cellules correspondant au logo dans l'image initiale. Soit x le nombre de cellules correspondant au logo dans votre solution. Le score pour ce test (sur 100) sera calculé selon la formule :

$$100 \times \left(\frac{x - b}{a - b} \right)^6$$

Par exemple, supposons que la solution optimale a $a = 100$ cellules correspondant au logo et que l'image initiale en a $b = 50$. Alors les scores seront attribués de la façon suivante :

Cellules correspondant au logo (x)	40	50	60	70	80	85	90	95	100
Score (% , arrondi)	0	0	0	0	5	12	26	53	100

La solution des juges trouve une solution optimale dans les limites de temps et de mémoire.

Le tableau suivant donne les tailles des entrées utilisées dans les fichiers tests.

R	C	Pourcentage des points
1	100	10%
10	10	10%
20	20	10%
100	100	70%