

BIFRÖST

Heimdall est en train de préparer son voyage annuel le long du *Bifröst*, le chemin arc-en-ciel vers les cieux. Le chemin consiste en k villes, et les villes consécutives sont reliés par des *ponts*. Chaque pont est d'une certaine couleur, représentée par un entier positif entre 0 et 5 000 inclus. Les $k - 1$ ponts sont numérotés de 1 à $k - 1$ de la gauche vers la droite, et le i ème pont est de couleur c_i .

Le voyage d'Heimdall commence dans la ville la plus à gauche et se termine dans la ville la plus à droite, possiblement en repassant plusieurs fois par les même villes. Au total, il a traversé des ponts A fois. Le i ème pont qu'il a traversé était de couleur a_i , et il a noté cette information dans son carnet.

Par exemple, s'il y a $k = 8$ villes et que les ponts sont colorés selon 1, 3, 2, 3, 4, 4, 5, alors les couleurs écrites dans son carnet peuvent être :

- 1, 3, 2, 3, 4, 4, 5.
- 1, 3, 3, 1, 1, 3, 2, 3, 4, 4, 5.
- 1, 3, 2, 3, 4, 4, 4, 4, 3, 3, 4, 4, 5, 5, 5.

Ces possibilités sont montrées dans la figure ci-dessous.

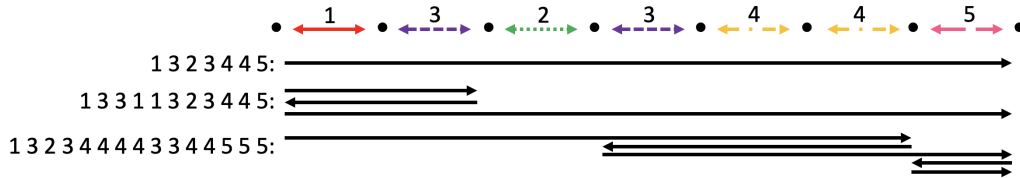


Figure 1: Trois voyages possibles. La couleur de chaque pont est notée au-dessus.

Cette année, Loki a fait le voyage entre la ville la plus à gauche et la ville la plus à droite. Au total, il a traversé des ponts B fois, le i ème étant de couleur b_i . Il a noté cela dans son carnet.

Heimdall pense que Loki ment peut-être, donc il vous a demandé de déterminer s'il existe un entier k et une suite $c_1, c_2, \dots, c_{k-1}$ qui correspond à la fois à son carnet et à celui de Loki.

Sous-tâches et contraintes

Pour chaque sous-tâche :

- $1 \leq A, B \leq 5\,000$.
- $0 \leq a_i, b_i \leq 5\,000$ pour tout i .

Additional constraints for each subtask are given below.

Sous-tâche	Points	Contraintes supplémentaires
1	6	$a_i = 0$ et $b_i = 0$ pour tout i .
2	10	Chaque couleur est présente au plus une fois dans le carnet d'Heimdall.
3	16	Chaque couleur (sauf 0) est présente au plus une fois dans le carnet d'Heimdall.
4	17	$A, B \leq 15$
5	34	$A, B \leq 100$
6	17	$A, B \leq 5000$

Entrée

- La première ligne de l'entrée contient les entiers A et B .
- La deuxième ligne contient A entiers $a_1, a_2, \dots, a_A$.
- La troisième ligne contient B entiers $b_1, b_2, \dots, b_B$.

Sortie

S'il n'existe aucune séquence qui correspond à la fois aux carnets d'Heimdall et de Loki, affichez une unique ligne IMPOSSIBLE.

Sinon, affichez deux lignes :

- Sur la première ligne, affichez k , le nombre de villes. Votre choix de k doit être au plus 5001.
- Sur la deuxième ligne, affichez $k - 1$ entiers $c_1, c_2, \dots, c_{k-1}$.

S'il y a plusieurs séquences correctes, vous pouvez afficher n'importe laquelle. Vous **n'avez pas** besoin de minimiser ou maximiser k .

Sample Input 1

```
11 15
1 3 3 1 1 3 2 3 4 4 5
1 3 2 3 4 4 4 3 3 4 4 5 5 5
```

Sample Output 1

```
8
1 3 2 3 4 4 5
```

Sample Input 2

```
4 3
5 0 0 3
5 0 3
```

Sample Output 2

```
IMPOSSIBLE
```

Explication

L'exemple d'entrée 1 correspond au cas montré dans la Figure 1. Une solution possible contient $k = 8$ villes, avec les $k - 1$ ponts colorés selon 1, 3, 2, 3, 4, 4, 5 de la gauche vers la droite. Ainsi, Heimdall peut avoir fait le voyage suivant :

- Il commence par traverser le pont 1 vers la droite (de couleur 1),
- Ensuite, il traverse le pont 2 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 2 vers la gauche (de couleur 3),
- Ensuite, il traverse le pont 1 vers la gauche (de couleur 1),
- Ensuite, il traverse le pont 1 vers la droite (de couleur 1),
- Ensuite, il traverse le pont 2 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 3 vers la droite (de couleur 2),
- Ensuite, il traverse le pont 4 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 5 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 6 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 7 vers la droite (de couleur 5), et finit son voyage.

Loki peut avoir fait le voyage suivant :

- Il commence par traverser le pont 1 vers la droite (de couleur 1),
- Ensuite, il traverse le pont 2 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 3 vers la droite (de couleur 2),
- Ensuite, il traverse le pont 4 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 5 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 6 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 6 vers la gauche (de couleur 4),
- Ensuite, il traverse le pont 5 vers la gauche (de couleur 4),
- Ensuite, il traverse le pont 4 vers la gauche (de couleur 3),
- Ensuite, il traverse le pont 4 vers la droite (de couleur 3),
- Ensuite, il traverse le pont 5 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 6 vers la droite (de couleur 4),
- Ensuite, il traverse le pont 7 vers la droite (de couleur 5),
- Ensuite, il traverse le pont 7 vers la gauche (de couleur 5),
- Ensuite, il traverse le pont 7 vers la droite (de couleur 5), et finit son voyage.

Dans l'exemple d'entrée 2, il n'existe pas de séquence qui corresponde à la fois au carnet d'Heimdall et à celui de Loki.

BONGI

Arrêtez tout ce que vous faites, un nouveau jeu super amusant vient juste de sortir : *Bongi*. Ce jeu est joué avec N balles. La i ème balle a l'entier a_i inscrit dessus.

Vous devez séparer les balles en K paniers, numérotés de 1 à K . Chaque balle doit aller dans un des paniers, et chaque panier doit contenir au moins une balle. La *valeur* d'un panier est la somme des entiers écrits sur les balles qui sont dedans. Votre *score* est la moyenne géométrique¹ des valeurs de tous les paniers.

Dans l'exemple suivant avec $K = 2$ paniers et $N = 5$ balles de valeurs 12, 14, 16, 17 et 25, une solution possible est la suivante :

- Le premier panier contient 16 et 17, et
- Le deuxième panier contient 25, 12 et 14.

Le score de cette solution est $(16 + 17)^{0.5} \cdot (25 + 12 + 14)^{0.5} \approx 41.02$.

Vous n'avez pas à donner le score maximal possible, en revanche le nombre de point qui vous sera accordés pour cet exercice dépendra du score maximal que vous êtes capable de produire.

C'est un exercice de **sortie uniquement**. Vous ne devez pas soumettre votre code source pour cet exercice. À la place, il vous est donné un ensemble de fichiers d'entrée et vous devez soumettre les fichiers de sortie correspondants.

Contraintes

Pour tous les tests :

- $1 \leq a_i \leq 10000$ pour tout i .
- $2 \leq K \leq N \leq 100$.

Entrée

- La première ligne de l'entrée contient les deux entiers N et K .
- La deuxième ligne contient N entiers $a_1, a_2, \dots, a_N$.

Sortie

Affichez K lignes, chacune contenant une collection d'entiers, telles que la i ème ligne contienne les valeurs des balles dans le i ème panier.

¹La moyenne géométrique de k valeurs $a_1, a_2, \dots, a_k$ est définie comme $a_1^{1/k} \times a_2^{1/k} \times \dots \times a_k^{1/k}$. Par exemple, la moyenne géométrique de 8, 9 et 10 est $8^{1/3} \times 9^{1/3} \times 10^{1/3} \approx 2 \times 2.080 \times 2.154 \approx 8.963$.

Score

Si vous ne mettez pas chaque balle dans exactement un panier, et que tous vos paniers ne contiennent pas au moins une balle, vous aurez 0% des points pour chaque test.

Sinon, si X est le score de votre solution, B celui de la meilleure solution des juges et $S = a_1 + a_2 + \dots + a_N$. Alors votre score au test sera

$$\min\left(100 \cdot \frac{10 + S/K - B}{10 + S/K - X}, 100\right)$$

pourcents des points de ce test.

Notez qu'il est possible de prouver que $X \leq S/K$, ainsi la fonction de score est bien définie.

Test	Points	N	K	Meilleur score des juges
1	5	5	2	42
2	10	20	10	9564.453334019241
3	10	100	10	55237.69999809915
4	15	100	20	25324.1999731493
5	20	100	30	16980.161334096203
6	20	100	40	13886.860798648839
7	10	100	50	10086.367502580433
8	10	100	60	8392.505093895417

Soumission

Pour soumettre votre fichier de sortie pour chaque test X , vous devez déposer un fichier nommé `output_X.txt`. Par exemple, pour le test 4, votre fichier de sortie doit être nommé `output_4.txt`.

Vous pouvez choisir de déposer un fichier de sortie pour chaque test séparément, ou tous ensemble dans un fichier zip.

Votre score à ce problème est la somme de votre score maximal à chacun des tests, parmi toutes les soumissions que vous avez faites pour ce problème.

Sample Input 1

```
5 2
16 25 17 14 12
```

Sample Output 1

```
16 17
25 12 14
```

Explication

Le score de la solution est $X = (16 + 17)^{0.5} \cdot (25 + 12 + 14)^{0.5} \approx 41.02$. Pour cet exemple, la meilleur solution des juges est 42. Ainsi, le pourcentage de points obtenus par cette solution est

$$\approx 100 \cdot \frac{10 + 84/2 - 42}{10 + 84/2 - 41.02} \approx 91.11.$$

Cet exemple correspond au premier test, qui vaut 5 points. Le nombre de points obtenus par cette solution est donc $4.56/5$.

Exemple de code

Deux fichiers vous sont fournis, `template.cpp` et `bongi.hpp`, pour vous aider à implémenter votre solution.

Le fichier `bongi.hpp` contient deux fonctions qui peuvent vous être utiles :

```
double get_score(vector<vector<int> > solution);
```

Cette fonction retourne le score d'une solution. Par exemple, si `solution = [[ 16, 17 ], [ 25, 12, 14 ]]`, alors `get_score(solution)` renvoie (approximativement) 41.02.

```
double get_percentage(vector<vector<int> > solution);
```

Cette fonction retourne le pourcentages de points qu'obtiendrait une solution. Par exemple, si `solution = [[ 16, 17 ], [ 25, 12, 14 ]]`, alors `get_percentage(solution)` renvoie (approximativement) 91.11.

Le fichier `template.cpp` contient un exemple de `get_percentage`. Vous pouvez compiler `template.cpp` en utilisant cette commande :

```
g++ -Wall -std=c++17 -O2 -o template template.cpp
```

LABYRINTHE VÉGÉTAL

Puss et Kitty doivent faire face à la diabolique énigme du *Labyrinthe Végétal Bas-Droite*. Le labyrinthe est une grille de cases de taille $R \times C$ avec R lignes (numérotés de 1 à R du haut vers le bas) et C colonnes (numérotés de 1 à C de la gauche vers la droite). La case dans la r ème colonne et c ème ligne est notée (r, c) . Il y a trois types de cases, désignées par un caractère minuscule :

- D : à partir de cette case, le seul mouvement autorisé est un déplacement vers le bas (s'il existe une case qui permet le mouvement).
- R : à partir de cette case, le seul mouvement autorisé est un déplacement vers la droite (s'il existe une case qui permet le mouvement).
- B : à partir de cette case, il est possible de se déplacer vers le bas ou vers la droite (s'il existe des cases qui permettent le mouvement).

Il est possible d'atteindre la case (R, C) depuis n'importe quelle autre case du labyrinthe.

D	B	B	R	D	D	B	D
R	D	D	R	R	R	R	B
B	R	D	B	B	D	B	B
B	D	D	D	R	R	D	D
R	B	B	R	R	B	B	D

Figure 1: Le labyrinthe de l'exemple d'entrée 1. Les lignes noires épaisses sont tracées entre les cases où le déplacement est interdit.

Pour vaincre l'énigme, Puss et Kitty doivent répondre à Q requêtes. Dans chaque requête, il vous est donné quatre entiers a_i, b_i, c_i et d_i et vous devez répondre à la question suivante : Si Puss démarre à la case (a_i, b_i) et Kitty démarre à la case (c_i, d_i) , quel est le plus petit nombre total de mouvements à effectuer pour les faire se rejoindre sur une même case ? Puss et Kitty démarrent dans des cases différentes.

Sous-tâches et contraintes

Pour toutes les sous-tâches :

- $2 \leq R \leq 1000$
- $2 \leq C \leq 500\,000$
- $4 \leq R \times C \leq 1\,000\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq a_i \leq R$ et $1 \leq b_i \leq C$ pour tout i .
- $1 \leq c_i \leq R$ et $1 \leq d_i \leq C$ pour tout i .
- $(a_i, b_i) \neq (c_i, d_i)$ pour tout i .

Additional constraints for each subtask are given below.

Sous-tâche	Points	Contraintes supplémentaires
1	9	$R, C \leq 20$
2	17	$(a_i, b_i) = (1, 1)$ pour tout i .
3	20	Chaque case sera de type D ou R.
4	29	Il existe une suite de mouvements valides depuis $(1, 1)$ jusqu'à n'importe quelle case du labyrinthe.
5	25	No additional constraints.

Entrée

- La première ligne de l'entrée contient trois entiers R, C, Q .
- Les R lignes suivantes contiennent chacune une chaîne de C caractères, décrivant le labyrinthe. Le c ème caractère de la r ème ligne représente le type de la case (r, c) .
- Les Q lignes suivantes décrivent les requêtes. La i ème ligne contient les quatre entiers a_i, b_i, c_i et d_i .

Sortie

Affichez Q lignes. Sur la i ème ligne, affichez la réponse à la i ème requête.

Sample Input 1

```
5 8 3
DBBRDDBD
RDDRRRRB
BRDBBDBB
BDDDRRDD
RBBRRBBD
1 2 3 7
3 3 1 1
3 6 4 4
```

Sample Output 1

```
9
4
7
```

Sample Input 2

```
4 4 4
BBRD
DRBB
BDRD
BBRB
2 1 3 3
1 3 2 2
3 3 1 3
4 1 3 2
```

Sample Output 2

```
7
4
4
2
```

Explication

Dans l'exemple d'entrée 1 :

- Dans la première requête, Puss et Kitty démarrent aux cases (1,2) et (3,7) respectivement. Il est possible de les faire se rejoindre à la case (3,8), nécessitant 8 and 1 mouvements respectivement, pour un total de 9 mouvements.
- Dans la deuxième requête, Puss et Kitty démarrent aux cases (3,3) et (1,1) respectivement. Il est possible de les faire se rejoindre à la case (3,3), nécessitant 0 and 4 mouvements respectivement, pour un total de 4 mouvements.
- Dans la troisième requête, Puss et Kitty démarrent aux cases (3,6) et (4,4) respectivement. Il est possible de les faire se rejoindre à la case (5,7), nécessitant 3 and 4 mouvements respectivement, pour un total de 7 mouvements.

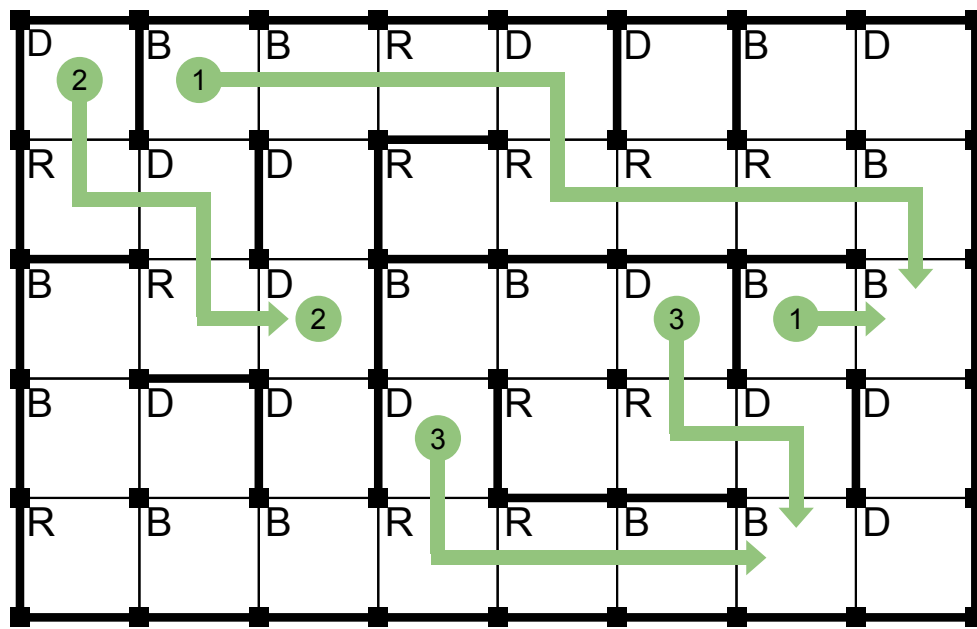


Figure 2: Exemple d'entrée 1

Dans l'exemple d'entrée 2 :

- Dans la première requête, Puss et Kitty démarrent aux cases (2, 1) et (3, 3) respectivement. Il est possible de les faire se rejoindre à la case (4, 4), nécessitant 5 and 2 mouvements respectivement, pour un total de 7 mouvements.
- Dans la deuxième requête, Puss et Kitty démarrent aux cases (1, 3) et (2, 2) respectivement. Il est possible de les faire se rejoindre à la case (2, 4), nécessitant 2 and 2 mouvements respectivement, pour un total de 4 mouvements.
- Dans la troisième requête, Puss et Kitty démarrent aux cases (3, 3) et (1, 3) respectivement. Il est possible de les faire se rejoindre à la case (3, 4), nécessitant 1 and 3 mouvements respectivement, pour un total de 4 mouvements.
- Dans la quatrième requête, Puss et Kitty démarrent aux cases (4, 1) et (3, 2) respectivement. Il est possible de les faire se rejoindre à la case (4, 2), nécessitant 1 and 1 move respectivement, pour un total de 2 mouvements.

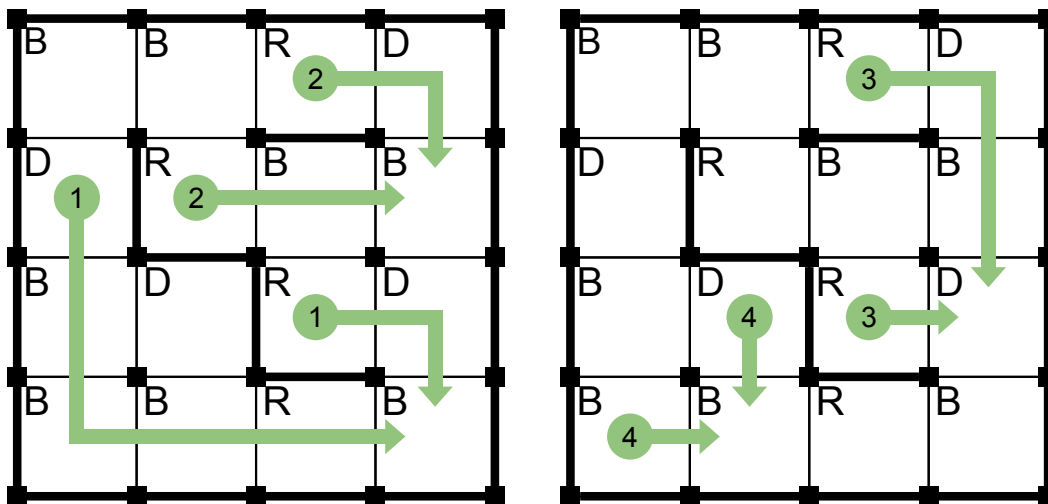


Figure 3: Exemple d'entrée 2. Les requêtes 3 et 4 sont affichées sur des copies différentes du labyrinthe pour plus de visibilité.