

Problem S1: Surmising a Sprinter's Speed

Problem Description

Trick E. Dingo is trying, as usual, to catch his nemesis the Street Sprinter. His past attempts using magnets, traps and explosives have failed miserably, so he's catching his breath to gather observational data and learn more about how fast Street Sprinter is.

Trick E. Dingo and Street Sprinter both inhabit a single straight west-east road with a particularly famous rock on it known affectionately as The Origin. Positions on this straight road are measured numerically according to the distance from The Origin, and using negative numbers for positions west of The Origin and positive numbers for positions east of The Origin.

The observations by Trick E. Dingo each contain two numbers: a time, and the value of Street Sprinter's position on the road at that time. Given this information, what speed must Street Sprinter must be capable of?

Input Specification

The first line contains a number $2 \leq N \leq 100\,000$, the number of observations that follow. The next N lines each contain an integer $0 \leq T \leq 1\,000\,000\,000$ indicating the time, in seconds, of when a measurement was made, and an integer $-1\,000\,000\,000 \leq X \leq 1\,000\,000\,000$ indicating the position, in metres, of the Street Sprinter at that time. No two lines will have the same value of T .

For 7 of the 15 available marks, $N \leq 1000$.

Output Specification

Output a single number X , such that we can conclude that Street Sprinter's speed was at least X metres/second at some point in time, and such that X is as large as possible. If the correct answer is C , the grader will view X as correct if $|X - C|/C < 10^{-5}$.

Sample Input 1

```
3
0 100
20 50
10 120
```

Output for Sample Input 1

```
7.0
```

Explanation of Output for Sample Input 1

Since the Street Sprinter ran from position 100 to position 120 between time 0 and time 10, we know its speed must have been at least 2 at some point in time: if it was always less than 2, then the distance of 20 could not be covered in 10 seconds. Likewise, the speed must have been at least 7 in order to travel between position 120 and 50 in 10 seconds.

La version française figure à la suite de la version anglaise.

Sample Input 2

5
20 -5
0 -17
10 31
5 -3
30 11

Output for Sample Input 2

6.8

Problème S1 : Supposer la vitesse d'un Sprinteur

Énoncé du problème

Hugo le Dingo tente, comme d'habitude, d'attraper son ennemi juré le Sprinteur Redoutable. Hélas toutes ses tentatives, orchestrées à l'aide d'aimants, de pièges et d'explosifs, ont abouti à un échec. Il décide donc de reprendre son haleine, tout en observant attentivement son adversaire redoutable, afin de recueillir des données quant à sa vitesse.

Hugo le Dingo et le Sprinteur Redoutable habitent tous deux sur une seule route allant de l'ouest à l'est et sur laquelle est situé un rocher particulièrement célèbre connu sous le nom de l'Origine. Les positions sur cette route droite sont mesurées numériquement en fonction de la distance de l'origine en utilisant des nombres négatifs pour les positions à l'ouest de l'origine et des nombres positifs pour les positions à l'est de l'origine.

À chaque fois que Hugo le Dingo recueille des données, il inscrit deux nombres : le premier d'eux représente un point dans le temps et le second représente la valeur de la position du Sprinteur Redoutable à ce moment-là. Compte tenu de ces informations, quelle est la vitesse minimale possible dont est capable le Sprinteur Redoutable ?

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un nombre $2 \leq N \leq 100\,000$, soit le nombre de fois que Hugo le Dingo recueille des données en observant le Sprinteur Redoutable. Les N lignes suivantes contiennent chacune un entier $0 \leq T \leq 1\,000\,000\,000$ représentant le point dans le temps, en secondes, où la donnée fut recueillie, et un entier $-1\,000\,000\,000 \leq X \leq 1\,000\,000\,000$ indiquant la position, en mètres, du Sprinteur Redoutable à ce moment précis. Aucune deux lignes ne contiendront la même valeur de T .

Pour 7 des 15 points disponibles, $N \leq 1000$.

Précisions par rapport aux données de sortie

Les données de sortie ne contiendront qu'un seul nombre, X , qui sera aussi grand que possible et qui sera tel que l'on puisse conclure que la vitesse du Sprinteur Redoutable était d'au moins X mètres/seconde à un moment donné. Si la bonne réponse est C , le correcteur considérera X comme étant la bonne réponse si $|X - C|/C < 10^{-5}$.

Données d'entrée d'un 1^{er} exemple

```
3
0 100
20 50
10 120
```

Données de sortie du 1^{er} exemple

```
7.0
```

Justification des données de sortie du 1^{er} exemple

Entre temps 0 et temps 10, le Sprinteur Redoutable a couru de la position 100 à la position 120, on sait donc que sa vitesse a dû être d'au moins 2 à un moment donné : afin de parcourir une distance de 20 en l'espace de 10 secondes, il aurait fallu qu'il atteigne une vitesse minimale d'au moins 2 car sinon il n'aurait pas pu parcourir toute cette distance en cet espace de temps. De même, afin de courir de la position 120 à la position 50 en l'espace de 10 secondes, il aurait fallu qu'il atteigne une vitesse minimale d'au moins 7.

Données d'entrée d'un 2^e exemple

5
20 -5
0 -17
10 31
5 -3
30 11

Données de sortie du 2^e exemple

6.8

Problem J5/S2: Escape Room

Problem Description

You have to determine if it is possible to escape from a room. The room is an M -by- N grid with each position (cell) containing a positive integer. The rows are numbered $1, 2, \dots, M$ and the columns are numbered $1, 2, \dots, N$. We use (r, c) to refer to the cell in row r and column c .

You start in the top-left corner at $(1, 1)$ and exit from the bottom-right corner at (M, N) . If you are in a cell containing the value x , then you can jump to any cell (a, b) satisfying $a \times b = x$. For example, if you are in a cell containing a 6, you can jump to cell $(2, 3)$.

Note that from a cell containing a 6, there are up to four cells you can jump to: $(2, 3)$, $(3, 2)$, $(1, 6)$, or $(6, 1)$. If the room is a 5-by-6 grid, there isn't a row 6 so only the first three jumps would be possible.

Input Specification

The first line of the input will be an integer M ($1 \leq M \leq 1000$). The second line of the input will be an integer N ($1 \leq N \leq 1000$). The remaining input gives the positive integers in the cells of the room with M rows and N columns. It consists of M lines where each line contains N positive integers, each less than or equal to 1 000 000, separated by single spaces.

For 1 of the 15 available marks, $M = 2$ and $N = 2$.

For an additional 2 of the 15 available marks, $M = 1$.

For an additional 4 of the 15 available marks, all of the integers in the cells will be unique.

For an additional 4 of the 15 available marks, $M \leq 200$ and $N \leq 200$.

Output Specification

Output `yes` if it is possible to escape from the room. Otherwise, output `no`.

Sample Input

```
3
4
3 10 8 14
1 11 12 12
6 2 3 9
```

Output for Sample Input

```
yes
```

Explanation of Output for Sample Input

Starting in the cell at (1, 1) which contains a 3, one possibility is to jump to the cell at (1, 3). This cell contains an 8 so from it, you could jump to the cell at (2, 4). This brings you to a cell containing 12 from which you can jump to the exit at (3, 4). Note that another way to escape is to jump from the starting cell to the cell at (3, 1) to the cell at (2, 3) to the exit.

Notes

1. The online grader begins by testing submissions using the sample input. All other tests are skipped if the sample test is not passed. If you are only attempting the first three subtasks (the first 7 marks), then you might want to handle the specific values of the sample input as a special case.
2. For the final subtask (worth 2 marks), if you are using Java, then `Scanner` will probably take too long to read in the large amount of data. A much faster alternative is `BufferedReader`.

Problème J5/S2: Le jeu d'évasion

Énoncé du problème

Vous devez déterminer s'il est possible de s'échapper d'une salle. La salle est en forme de grille $M \times N$. Chacune des cases de la grille contient un entier strictement positif. Les rangées de la grille sont numérotées $1, 2, \dots, M$ tandis que les colonnes sont numérotées $1, 2, \dots, N$. Les coordonnées (r, c) représentent la case qui est située dans la rangée r de la colonne c .

Vous commencez le jeu à partir de la case $(1, 1)$ au coin supérieur gauche de la grille et vous vous échappez de la salle en atteignant la case (M, N) au coin inférieur droit de la grille. Si vous trouvez sur une case contenant la valeur x , alors vous pouvez accéder à n'importe quelle case (a, b) telle que $a \times b = x$. Par exemple, une case contenant un 6 vous donne accès à la case $(2, 3)$.

On remarque qu'une case contenant un 6 nous donne accès à quatre cases, soient les cases $(2, 3)$, $(3, 2)$, $(1, 6)$ et $(6, 1)$. Si la salle est en forme de grille 5×6 , il n'y a pas de 6^e rangée, donc l'accès ne sera limité qu'aux trois premières cases, soient les cases $(2, 3)$, $(3, 2)$ et $(1, 6)$.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un entier M qui vérifie $1 \leq M \leq 1000$. La deuxième ligne des données d'entrée contient un entier N qui vérifie $1 \leq N \leq 1000$. Le restant des données d'entrée contient les entiers strictement positifs dans les cases de la salle à M rangées et à N colonnes. On s'attend donc à ce que le restant des données d'entrée contienne M lignes et que chaque ligne contienne N entiers strictement positifs chacun étant séparé des autres par un espace; chacun de ces entiers étant au plus égal à 1 000 000.

Pour 1 des 15 points disponibles, $M = 2$ et $N = 2$.

Pour 2 autres points parmi les 15 points disponibles, $M = 1$.

Pour 4 autres points parmi les 15 points disponibles, les entiers dans les cases seront tous différents les uns des autres.

Pour 4 autres points parmi les 15 points disponibles, $M \leq 200$ et $N \leq 200$.

Précisions par rapport aux données de sortie

Les données de sortie devraient afficher `yes` s'il est possible de s'échapper de la salle, sinon elles devraient afficher `no`.

Exemple de données d'entrée

```
3
4
3 10 8 14
1 11 12 12
6 2 3 9
```

Exemple de données de sortie

yes

Justification des données de sortie

En commençant par la case (1, 1) qui contient l'entier 3, on peut emprunter le chemin (1, 1) → (1, 3) → (2, 4) → (3, 4) afin de s'échapper de la salle selon le raisonnement suivant: l'entier 3 dans la case (1, 1) nous donne accès à la case (1, 3), l'entier 8 dans la case (1, 3) nous donne accès à la case (2, 4), l'entier 12 dans la case (2, 4) nous donne accès à la case (3, 4) d'où on peut s'échapper de la salle. Remarquons aussi qu'on pourrait s'échapper de la salle en empruntant le chemin suivant: (1, 1) → (3, 1) → (2, 3) → (3, 4).

Notes

1. Le correcteur informatisé commence par tester les soumissions à l'aide de l'échantillon d'entrée. Si l'échantillon testé n'est pas accepté, alors tous les autres tests sont ignorés. Ainsi, si vous ne tentez que les trois premières sous-tâches (les 7 premiers points), ensuite vous pourriez vouloir traiter les valeurs spécifiques de l'échantillon d'entrée en tant que cas particulier.
2. Pour la dernière sous-tâche (qui vaut 2 points), si vous utilisez Java, alors le `Scanner` va probablement prendre trop de temps pour lire la masse de données. Une alternative beaucoup plus rapide est le `BufferedReader`.

Problem S3: Searching for Strings

Problem Description

You're given a string N , called the *needle*, and a string H , called the *haystack*, both of which contain only lowercase letters "a".."z".

Write a program to count the number of distinct permutations of N which appear as a substring of H at least once. Note that N can have anywhere between 1 and $|N|!$ distinct permutations in total – for example, the string "aab" has 3 distinct permutations ("aab", "aba", and "baa").

Input Specification

The first line contains N ($1 \leq |N| \leq 200\,000$), the needle string.

The second line contains H ($1 \leq |H| \leq 200\,000$), the haystack string.

For 3 of the 15 available marks, $|N| \leq 8$ and $|H| \leq 200$.

For an additional 2 of the 15 available marks, $|N| \leq 200$ and $|H| \leq 200$.

For an additional 2 of the 15 available marks, $|N| \leq 2000$ and $|H| \leq 2000$.

Output Specification

Output consists of one integer, the number of distinct permutations of N which appear as a substring of H .

Sample Input

```
aab
abacabaa
```

Output for Sample Input

```
2
```

Explanation of Output for Sample Input

The permutations "aba" and "baa" each appear as substrings of H (the former appears twice), while the permutation "aab" does not appear.

Problème S3 : À la recherche de chaînes

Énoncé du problème

Soit N une chaîne appelée *Aiguille* et H une chaîne appelée *Botte de foin*, les deux ne contenant comme caractères que des lettres minuscules « a »..« z ».

Écrire un programme dont le but est de compter le nombre d'arrangements distincts de N qui paraissent au moins une seule fois comme sous-chaînes de H . Remarquons que N peut avoir un nombre total d'arrangements distincts allant de 1 à $|N|!$ arrangements. Par exemple, la chaîne « aab » a 3 arrangements distincts (« aab », « aba » et « baa »).

Précisions par rapport aux données d'entrée

La première ligne contient N ($1 \leq |N| \leq 200\,000$), soit la chaîne *Aiguille*.

La deuxième ligne contient H ($1 \leq |H| \leq 200\,000$), soit la chaîne *Botte de foin*.

Pour 3 des 15 points disponibles, $|N| \leq 8$ et $|H| \leq 200$.

Pour 2 autres points parmi les 15 points disponibles, $|N| \leq 200$ et $|H| \leq 200$.

Pour 2 autres points parmi les 15 points disponibles, $|N| \leq 2000$ et $|H| \leq 2000$.

Précisions par rapport aux données de sortie

Les données de sortie ne devraient contenir qu'un seul entier, soit le nombre d'arrangements distincts de N qui paraissent comme sous-chaînes de H .

Exemple de données d'entrée

```
aab
abacabaa
```

Exemple de données de sortie

```
2
```

Justification des données de sortie

Les arrangements « aba » et « baa » paraissent tous les deux comme sous-chaînes de H (on voit d'ailleurs que le premier y paraît deux fois) tandis que l'arrangement « aab » n'y paraît aucunement.

Problem S4: Swapping Seats

Problem Description

There are N people sitting at a circular table for a long session of negotiations. Each person belongs to one of the three groups: A, B, or C. A group is *happy* if all of its members are sitting contiguously in a block of consecutive seats. You would like to make all groups happy by some sequence of *swap* operations. In each swap operation, two people exchange seats with each other. What is the minimum number of swaps required to make all groups happy?

Input Specification

The input consists of a single line containing N ($1 \leq N \leq 1\,000\,000$) characters, where each character is A, B, or C. The i -th character denotes the group of the person initially sitting at the i -th seat at the table, where seats are numbered in clockwise order.

For 4 of the 15 available marks, the input has no C characters and $N \leq 5\,000$.

For an additional 4 of the 15 available marks, the input has no C characters.

For an additional 4 of the 15 available marks, $N \leq 5\,000$.

Output Specification

Output a single integer, the minimum possible number of swaps.

Sample Input

BABCBCACCA

Output for Sample Input

2

Explanation of Output for Sample Input

In one possible sequence, the first swap results in the seating layout AABCBCBCCA. After the second swap, the layout is AABBBCCCCA.

Problème S4 : Un échange de places

Énoncé du problème

Il y a N personnes assises autour d'une table circulaire pendant une longue période de négociations. Chaque personne appartient à un des trois groupes suivants : A, B ou C. Le groupe est *heureux* lorsque tous ses membres sont assis côte à côte dans un bloc de chaises consécutives. Il faut tenter de rendre tous les groupes heureux par une série d'*échanges*. Lors d'un échange, deux personnes sont choisies au hasard et elles échangent leur place l'une avec l'autre. Quel est le nombre minimal d'échanges requis afin de rendre tous les groupes heureux ?

Précisions par rapport aux données d'entrée

Les données d'entrée ne contiennent qu'une seule ligne. Cette ligne contient des caractères N ($1 \leq N \leq 1\,000\,000$) dont chacun est soit A, soit B, soit C. Le i -ième caractère désigne le groupe de la personne qui était initialement assise à la i -ième place à la table, où les places sont numérotées en ordre dans le sens des aiguilles d'une montre.

Pour 4 des 15 points disponibles, les données d'entrée ne contiennent aucun caractère C et $N \leq 5\,000$.

Pour 4 autres points parmi les 15 points disponibles, les données d'entrée ne contiennent aucun caractère C.

Pour 4 autres points parmi les 15 points disponibles, $N \leq 5\,000$.

Précisions par rapport aux données de sortie

Les données de sortie ne devraient contenir qu'un seul entier, soit le nombre minimal d'échanges requis.

Exemple de données d'entrée

BABCBCACCA

Exemple de données de sortie

2

Justification des données de sortie

Dans une série d'échanges possibles, on obtient AABCBCBCCA d'un premier échange et AABBBCCCCA d'un second.

Problem S5: Josh's Double Bacon Deluxe

Problem Description

On their way to the contest grounds, Josh, his coach, and his $N - 2$ teammates decide to stop at a burger joint that offers M distinct burger menu items. After ordering their favourite burgers, the team members line up, with the coach in the first position and Josh last, to pick up their burgers. Unfortunately, the coach forgot what he ordered. He picks a burger at random and walks away. The other team members, in sequence, pick up their favourite burger if available, or a random remaining burger if there are no more of their favourite burger. What is the probability that Josh, being last in line, will get to eat his favourite burger?

Input Specification

The first line contains the number N ($3 \leq N \leq 1\,000\,000$), the total number of people and burgers. The next line contains N numbers, the i -th being b_i ($1 \leq b_i \leq M \leq 500\,000$), denoting the item number of the i -th person's favourite burger. The first person in line is the coach, and the N -th person is Josh.

For 4 of the 15 available marks, $N \leq 100\,000$ and $M \leq 1000$

For an additional 5 of the 15 available marks, $M \leq 5000$.

Output Specification

Output a single number P , the probability that Josh will get to eat his favourite burger, b_N . If the correct answer is C , the grader will view P correct if $|P - C| < 10^{-6}$.

Sample Input 1

```
3
1 2 3
```

Output for Sample Input 1

```
0.5
```

Explanation of Output for Sample Input

The coach randomly chooses between the three burgers. With probability $1/3$, he chooses his favourite burger (burger 1), and Josh gets to eat his favourite burger (burger 3). With probability $1/3$, he chooses Josh's favourite burger, and Josh fails to eat his favourite burger. With probability $1/3$, he chooses the second person's burger, there is a $1/2$ chance that the second person chooses Josh's burger, denying Josh the pleasure of eating his favourite burger.

Sample Input 2

```
7
1 2 3 1 1 2 3
```

La version française figure à la suite de la version anglaise.

Output for Sample Input 2

0.57142857

La version française figure à la suite de la version anglaise.

Problème S5 : Jean et son burger au bacon

Énoncé du problème

En chemin vers la compétition, Jean, son entraîneur et $N - 2$ de ses concurrents décident de s'arrêter à un restaurant rapide qui offre M choix de burgers différents sur leur menu. Après avoir commandé leur burger préféré, les compétiteurs s'alignent avec leur entraîneur en première position et Jean en dernière pour récupérer leur commande. Malheureusement, l'entraîneur a oublié ce qu'il a commandé. Il récupère donc un burger au hasard et quitte le restaurant. Un par un, les autres compétiteurs prennent leur burger préféré ou tout autre burger restant au hasard si le leur n'est pas disponible, tout en respectant l'ordre de la file. Quelle est la probabilité que Jean, étant le dernier de la file, ait son burger préféré ?

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient le nombre N ($3 \leq N \leq 1\,000\,000$), soit le nombre total de personnes et de burgers. La ligne suivante contient N nombres, dont le i -ième étant b_i ($1 \leq b_i \leq M \leq 500\,000$). Ce dernier représente le numéro associé au choix de burger de la i -ième personne. Dans la file, l'entraîneur est en première position tandis que Jean est en N -ième position.

Pour 4 des 15 points disponibles, $N \leq 100\,000$ et $M \leq 1000$

Pour 5 autres points parmi les 15 points disponibles, $M \leq 5000$.

Précisions par rapport aux données de sortie

Les données de sortie ne devraient contenir qu'un seul nombre, P , soit la probabilité que Jean ait son burger préféré, b_N . Si la bonne réponse est C , le correcteur considérera P comme étant la bonne réponse si $|P - C| < 10^{-6}$.

Données d'entrée d'un 1^{er} exemple

3

1 2 3

Données de sortie du 1^{er} exemple

0.5

Justification des données de sortie du 1^{er} exemple

L'entraîneur choisit au hasard un burger parmi trois. Voici donc les différents résultats possibles :

- L'entraîneur choisit son burger préféré (le burger 1) et Jean reçoit son burger préféré (le burger 3). La probabilité de ce résultat est égale à $1/3$.
- L'entraîneur choisit le burger préféré de Jean et donc Jean ne reçoit pas son burger préféré. La probabilité de ce résultat est égale à $1/3$.
- L'entraîneur choisit le burger préféré de la 2^e personne dans la file. La probabilité de ce résultat est égale à $1/3$. De plus, après ce premier choix, il y a une probabilité de $1/2$ que

cette deuxième personne choisisse le burger préféré de Jean. Dans ce cas, Jean ne recevra pas son burger préféré.

Données d'entrée d'un 2^e exemple

7

1 2 3 1 1 2 3

Données de sortie du 2^e exemple

0.57142857