

Problem J1: Roller Coaster Ride

Problem Description

You are spending the day at the CEMC's funfair. One of the rides at the funfair is a roller coaster which has one train with a number of cars. Each car holds the same number of people.

When you arrive at the roller coaster, you see that there is a line. Your job is to determine whether or not you will be on the next train ride, assuming that every car is fully occupied for every ride.



Input Specification

The first line of input contains a positive integer, N , representing your place in line. For example, if N is 5 then you are the fifth person in line.

The second line contains a positive integer, C , representing the number of cars the train has.

The third line contains a positive integer, P , representing the number of people a single car holds.

Output Specification

Output either **yes** or **no**, indicating whether or not you will be on the next train ride.

Sample Input 1

14
3
2

Output for Sample Input 1

no

Explanation of Output for Sample Input 1

The train has 3 cars and each car holds 2 people. Therefore, 6 people can ride the next train. Since you are the 14th person in line, you will not be on the next train ride.

Sample Input 2

12
4
3

Output for Sample Input 2

yes

Explanation of Output for Sample Input 2

The train has 4 cars and each car holds 3 people. Therefore, 12 people can ride the next train. Since you are the 12th person in line, you will be on the next train ride.

Problème J1 : Montagnes russes

Énoncé du problème

Vous passez la journée à la fête foraine du CEMI. L'un des manèges à la fête foraine est une montagne russe avec un train composé de plusieurs wagons. Chaque wagon peut transporter le même nombre de personnes.

Lorsque vous arrivez au manège, vous constatez qu'il y a une file d'attente. Votre tâche consiste à déterminer si vous pourrez monter à bord du prochain train, en supposant que, tous les tours, tous les wagons sont entièrement occupés.



Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un entier strictement positif, N , représentant votre rang dans la ligne. Par exemple, si N est égal à 5, vous êtes la cinquième personne dans la file d'attente.

La deuxième ligne des données d'entrée contient un entier strictement positif, W , représentant le nombre de wagons du train.

La troisième ligne des données d'entrée contient un entier strictement positif, P , représentant le nombre de personnes dans un seul wagon.

Précisions par rapport aux données de sortie

Les données de sorties devraient être **yes** ou **no**, indiquant si vous serez ou non à bord du prochain train.

Données d'entrée d'un 1^{er} exemple

14
3
2

Données de sortie du 1^{er} exemple

no

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

Le train est composé de 3 wagons et chaque wagon contient 2 personnes. Par conséquent, 6 personnes monteront à bord du prochain train. Comme vous êtes la 14^e personne dans la file d'attente, vous ne pourrez pas monter à bord du prochain train.

Données d'entrée du 2^e exemple

12
4
3

Données de sortie du 2^e exemple

yes

Justification des données de sortie du 2^e exemple

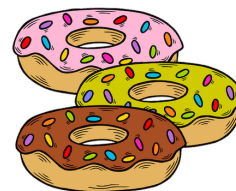
Le train est composé de 4 wagons et chaque wagon contient 3 personnes. Par conséquent, 12 personnes monteront à bord du prochain train. Comme vous êtes la 12^e personne dans la file d'attente, vous pourrez monter à bord du prochain train.

Problem J2: Donut Shop

Problem Description

The owner of a donut shop spends the day baking and selling donuts.

Given the events that happen over the course of the day, your job is to determine the number of donuts remaining when the shop closes.



Input Specification

The first line of input contains a non-negative integer, D , representing the number of donuts available when the shop first opens.

The second line contains a positive integer, E , representing the number of events that happen over the course of the day. The next E pairs of input lines describe these events.

The first line in the pair contains either the + (plus) symbol, indicating that donuts have been baked, or the - (minus) symbol, indicating that donuts have been sold. The second line in the pair contains a positive integer, Q , representing the quantity of donuts associated with the event.

For each sale of donuts, the value of Q will be less than or equal to the number of donuts available at that time.

Output Specification

Output the non-negative integer, R , which is the number of donuts remaining when the shop closes.

Sample Input

```
10
3
+
24
-
6
-
12
```

Output for Sample Input

```
16
```

Explanation of Output for Sample Input

The shop opened with 10 donuts and there were 3 events during the day. The owner first baked 24 donuts. Then the owner sold 6 donuts, followed by another 12. The number of donuts remaining is $10 + 24 - 6 - 12 = 16$.

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

Problème J2 : Beignerie

Énoncé du problème

La propriétaire d'une beignerie passe sa journée à préparer et à vendre des beignets.

Votre tâche consiste à déterminer le nombre de beignets invendus au moment de la fermeture de la beignerie, en vous basant sur les événements de la journée.



Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un entier positif, B , représentant le nombre de beignets en vente à l'ouverture de la beignerie.

La deuxième ligne des données d'entrée contient un entier strictement positif, E , représentant le nombre d'événements de la journée. Les E paires de lignes de données d'entrée suivantes décrivent ces événements.

La première ligne de chaque paire contient soit le symbole $+$ (plus), indiquant que des beignets ont été préparés, soit le symbole $-$ (moins), indiquant que des beignets ont été vendus. La deuxième ligne de la paire de lignes contient un entier strictement positif, Q , représentant la quantité de beignets associée à l'événement.

Pour chaque vente de beignets, la valeur de l'entier strictement positif Q sera inférieure ou égale au nombre de beignets disponibles à ce moment-là.

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir un entier positif, I , représentant le nombre de beignets invendus à la fermeture de la boutique.

Données d'entrée

```
10
3
+
24
-
6
-
12
```

Données de sortie

```
16
```

Justification des données de sortie

Lorsque la boutique a ouvert ses portes, 10 beignets étaient en vente, puis 3 événements ont eu lieu au cours de la journée. La propriétaire a d'abord préparé 24 beignets. Elle a ensuite vendu 6 beignets, puis 12 autres. Le nombre de beignets invendus au moment de la fermeture est de $10 + 24 - 6 - 12 = 16$.

Problem J3: Product Codes

Problem Description

A store has hired the Code Cleaning Crew to help it update all of its product codes.

The original product codes are sequences of letters, positive integers, and negative integers. For example, `cG23mH-9s` is a product code that contains two uppercase letters, three lowercase letters, one positive integer, and one negative integer.

The new product codes are made by removing all lowercase letters, keeping all uppercase letters in order, and adding all the integers to form one new integer which is placed at the end of the code. For example, the new product code for `cG23mH-9s` is `GH14`.

Your job is to take a list of original product codes and determine the new product codes.

Input Specification

The first line of input contains a positive integer, N , representing the number of original product codes that need to be updated. The following N lines each contain one original product code.

Each original product code contains at least one uppercase letter, at least one lowercase letter, and at least one integer. Also, a positive integer never immediately follows another integer. This means, for example, that `23` is the integer 23 instead of the integer 2 followed by the integer 3.

The following table shows how the available 15 marks are distributed:

Marks	Description
2	All the integers are positive and single-digit.
2	All the integers are single-digit.
7	Any positive integer may be multi-digit.
4	Any integer may be multi-digit.

Output Specification

Output the N new product codes, one per line.

Sample Input 1

```
1
AbC3c2Cd9
```

Output for Sample Input 1

```
ACC14
```

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

Explanation of Output for Sample Input 1

For the single original product code, the uppercase letters A, C, and C are kept in order and the sum of the integers is $3 + 2 + 9 = 14$.

Sample Input 2

```
3
Ahkiy-6ebvXCV1
393hhhUHkbs5gh6QpS-9-8
PL12N-2G1234Duytrty8-86tyaYySsDdEe
```

Output for Sample Input 2

```
AXCV-5
UHQS387
PLNGDYSDE1166
```

Explanation of Output for Sample Input 2

For the first original product code, the uppercase letters A, X, C, and V are kept in order and the sum of the integers is $-6 + 1 = -5$.

For the second and third original product codes, their uppercase letters are also kept in order and the sums of the integers are $393 + 5 + 6 - 9 - 8 = 387$ and $12 - 2 + 1234 + 8 - 86 = 1166$ respectively.

Problème J3 : Codes de produits

Énoncé du problème

Le gérant d'un magasin a embauché la Compagnie de correction d'identifiants (CCI) pour l'aider à mettre à jour tous ses codes de produits.

Les codes de produits originaux sont des séquences de lettres, d'entiers strictement positifs et d'entiers strictement négatifs. Par exemple, `cG23mH-9s` est un code de produit qui contient deux lettres majuscules, trois lettres minuscules, un entier strictement positif et un entier strictement négatif.

Les nouveaux codes de produits sont obtenus en éliminant toutes les lettres minuscules, en conservant toutes les lettres majuscules dans l'ordre et en additionnant tous les entiers pour former un nouveau nombre entier qui sera placé à la fin du code. Par exemple, le nouveau code de produit pour le code `cG23mH-9s` est : `GH14`.

Votre tâche consiste à prendre une liste de codes de produits originaux et à déterminer les nouveaux codes de produits.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient un entier strictement positif, N , représentant le nombre de codes de produits originaux devant être corrigés. Les N lignes de données d'entrée suivantes contiennent toutes un code de produit original.

Chaque code de produit original contient au moins une lettre majuscule, une lettre minuscule et un entier. Un entier strictement positif ne suit jamais immédiatement un autre entier. Par exemple, cela signifie que `23` correspond à l'entier 23 et non pas à l'entier 2 suivi de l'entier 3.

Le tableau suivant détaille la répartition des 15 points disponibles.

Points	Description
2	Tous les entiers sont strictement positifs et comportent un seul chiffre.
2	Tous les entiers comportent un seul chiffre.
7	Tout entier strictement positif peut comporter plusieurs chiffres.
4	Tout entier peut comporter plusieurs chiffres.

Précisions par rapport aux données de sortie

Les données de sorties devraient contenir les N nouveaux codes de produit, un par ligne.

Données d'entrée d'un 1^{er} exemple

1

AbC3c2Cd9

Données de sortie du 1^{er} exemple

ACC14

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

Pour le premier code de produit original, les lettres majuscules A, C, et C sont conservées dans l'ordre et la somme des nombres entiers est de $3 + 2 + 9 = 14$.

Données d'entrée du 2^e exemple

3

Ahkiy-6ebvXCV1

393hhhUHKbs5gh6QpS-9-8

PL12N-2G1234Duytrty8-86tyaYySsDdEe

Données de sortie du 2^e exemple

AXCV-5

UHQS387

PLNGDYSDE1166

Justification des données de sortie du 2^e exemple

Pour le premier code de produit original, les lettres majuscules A, X, C, et V sont conservées dans l'ordre et la somme des nombres entiers est de $-6 + 1 = -5$.

Pour le deuxième et le troisième code de produits originaux, les lettres majuscules sont également conservées dans l'ordre et les sommes des nombres entiers sont respectivement $393 + 5 + 6 - 9 - 8 = 387$ et $12 - 2 + 1234 + 8 - 86 = 1166$.

Problem J4: Sunny Days

Problem Description

There is a large amount of historical weather data for CEMCity. Each day in the data is listed as either a day with sunshine or a day with precipitation. Jeremy is interested in finding the record for the most consecutive days with sunshine. Unfortunately, the data is incorrect for exactly one day, but Jeremy doesn't know which day this is.



Your job is to help Jeremy determine the maximum possible number of consecutive days with sunshine.

Input Specification

The first line of input contains a positive integer, N , representing the number of days in the historical data. The following N lines each contain either the character **S** or the character **P**, representing a day with sunshine or a day with precipitation, respectively, in chronological order.

The following table shows how the available 15 marks are distributed:

Marks	Description	Bounds
2	There are not many days in the historical data. The data consists of a single block of all S 's followed by a single block of all P 's. One of these blocks may be empty.	$N \leq 1000$
4	There are not many days in the historical data. The data contains S 's and P 's possibly in mixed order.	$N \leq 1000$
9	There are possibly many days in the historical data.	$N \leq 500\,000$

Output Specification

Output the non-negative integer, M , which is the maximum possible number of consecutive days with sunshine.

Sample Input

```
8
P
S
P
S
S
P
P
S
```

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

Output for Sample Input

4

Explanation of Output for Sample Input

If the data is incorrect for the third day, then there was sunshine from the second day to the fifth day which is four consecutive days with sunshine. This is the maximum possible number of consecutive days with sunshine. That is, no matter which day the data is incorrect for, there were not five (or more) consecutive days of sunshine.

Problème J4 : Journées ensoleillées

Énoncé du problème

Il existe un vaste ensemble de données météorologiques historiques pour la municipalité de CEMIVille. Chaque jour est catégorisé dans cet ensemble de données comme une journée de soleil ou une journée de pluie. Jérémie souhaite trouver le record du plus grand nombre de journées de soleil consécutives. Malheureusement, la donnée pour l'une des journées est erronée et Jérémie ne sait pas de quel jour il s'agit.



Votre tâche consiste à aider Jérémie à déterminer le nombre maximal de jours de soleil consécutifs possible.

Précisions par rapport aux données d'entrée

La première ligne de données d'entrée contient un entier strictement positif, N , représentant le nombre de jours dans l'ensemble de données historique. Les N lignes de données d'entrée suivantes contiennent chacune le caractère **S** ou le caractère **P**, représentant respectivement un jour de soleil ou de pluie, en ordre chronologique.

Le tableau suivant détaille la répartition des 15 points disponibles.

Points	Description	Bornes
2	Il n'y a pas beaucoup de jours dans l'ensemble de données historiques. Les données seront constituées d'un seul bloc de S suivi d'un seul bloc de P et l'un de ces blocs peut être vide.	$N \leq 1000$
4	Il n'y a pas beaucoup de jours dans l'ensemble de données historiques. Les données contiennent des S et des P , possiblement dans un ordre mixte.	$N \leq 1000$
9	Il est possible qu'il y ait beaucoup de jours dans l'ensemble de données historiques.	$N \leq 500\,000$

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir un entier positif, M , représentant le nombre maximal de jours de soleil consécutifs possible.

Données d'entrée

8
P
S
P
S
S
P
P
S

Données de sortie

4

Justification des données de sortie

Si la donnée du troisième jour est erronée, le soleil aura été présent du deuxième au cinquième jour, soit quatre jours de soleil consécutifs. Il s'agit du nombre maximal de jours de soleil consécutifs possible. Autrement dit, quel que soit le jour pour lequel la donnée est incorrecte, il ne peut y avoir eu cinq (ou plus) jours de soleil consécutifs.

Problem J5: Connecting Territories

Problem Description

Ava is playing a strategic game on a grid of tiles. Each tile has an associated cost. When the costs of the tiles are read from left to right, starting with the first row, a repeating pattern of the first M positive integers in increasing order is revealed: $1, 2, 3, \dots, M, 1, 2, 3, \dots, M, 1, 2, 3, \dots$. In this pattern, M represents the maximum cost of a tile. In the grid of tiles shown, M is equal to 5.

1	2	3	4	5	1	2	3
4	5	1	2	3	4	5	1
2	3	4	5	1	2	3	4
5	1	2	3	4	5	1	2
3	4	5	1	2	3	4	5

Ava needs to purchase one tile in each row in order to make a connecting path from the upper territory (above the first row of tiles) to the lower territory (below the last row of tiles). The first tile purchased must be in the first row. Each subsequently purchased tile must share either an edge or a corner with the tile purchased previously. In the grid of tiles shown, the cost of Ava's connecting path is 9.

Upper Territory							
1	2	3	4	5	1	2	3
4	5	1	2	3	4	5	1
2	3	4	5	1	2	3	4
5	1	2	3	4	5	1	2
3	4	5	1	2	3	4	5
Lower Territory							

Given a grid of tiles, your job is to determine the minimum cost of a connecting path between the upper and lower territories.

Input Specification

The first line of input contains a positive integer, R , representing the number of rows in the grid. The second line contains a positive integer, C , representing the number of columns in the grid. The third line contains a positive integer, M where $M \leq 100\,000$, representing the maximum cost of a tile.

The following table shows how the available 15 marks are distributed:

Marks	Description	Bounds
3	There are two rows and at most ten columns.	$R = 2$ and $C \leq 10$
8	There are at most ten rows and at most ten columns.	$R \leq 10$ and $C \leq 10$
2	There are at most 100 rows and at most 100 columns.	$R \leq 100$ and $C \leq 100$
2	The grid may be very large.	$R \leq 20\,000$ and $C \leq 20\,000$

Output Specification

Output the positive integer, P , which is the minimum cost of a connecting path between the upper and lower territories.

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

Sample Input

3
5
7

Output for Sample Input

6

Explanation of Output for Sample Input

The cost of each tile is shown. The sequence of tiles that Ava should purchase to minimize the cost of a connecting path is highlighted in green.

Upper Territory				
1	2	3	4	5
6	7	1	2	3
4	5	6	7	1
Lower Territory				

Problème J5 : Jonction de territoires

Énoncé du problème

Ève joue à un jeu de stratégie sur une grille de cases. Un coût est associé à chaque case. La lecture de gauche à droite et de haut en bas du coût des cases en commençant par la case en haut à gauche révèle une suite finie croissante des M premiers entiers strictement positifs qui se répète comme suit : $1, 2, 3, \dots, M, 1, 2, 3, \dots, M, 1, 2, 3, \dots$. Dans ce schéma, M représente le coût maximal d'une case. Dans l'illustration de la grille de cases, M est égal à 5.

1	2	3	4	5	1	2	3
4	5	1	2	3	4	5	1
2	3	4	5	1	2	3	4
5	1	2	3	4	5	1	2
3	4	5	1	2	3	4	5

Ève doit acheter une case dans chaque rangée afin de créer un chemin de jonction entre le territoire du haut (au-dessus de la première rangée de cases) et le territoire du bas (en dessous de la dernière rangée de cases). La première case achetée doit se trouver dans la première rangée. Ensuite, chaque case achetée doit partager un bord ou un coin avec la case précédente. Dans l'illustration de la grille de cases, le coût du chemin de jonction d'Ève est de 9.

Territoire du haut							
1	2	3	4	5	1	2	3
4	5	1	2	3	4	5	1
2	3	4	5	1	2	3	4
5	1	2	3	4	5	1	2
3	4	5	1	2	3	4	5
Territoire du bas							

À l'aide de la grille de cases, votre tâche consiste à déterminer le coût minimum d'un chemin de jonction entre le territoire du haut et le territoire du bas.

Précisions par rapport aux données d'entrée

La première ligne de données d'entrée contient un entier strictement positif, R , représentant le nombre de rangées de la grille. La deuxième ligne de données d'entrée contient un entier strictement positif, C , représentant le nombre de colonnes de la grille. La troisième ligne de données d'entrée contient un entier strictement positif, M ($M \leq 100\,000$), représentant le coût maximal d'une case.

Le tableau suivant détaille la répartition des 15 points disponibles.

Points	Description	Bornes
3	Il y a deux rangées et au plus dix colonnes.	$R = 2$ et $C \leq 10$
8	Il y a au plus dix rangées et au plus dix colonnes.	$R \leq 10$ et $C \leq 10$
2	Il y a au plus 100 rangées et au plus 100 colonnes.	$R \leq 100$ et $C \leq 100$
2	La grille peut être très grande.	$R \leq 20\,000$ et $C \leq 20\,000$

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir un entier strictement positif, P , représentant le coût minimum d'un chemin de jonction entre le territoire du haut et le territoire du bas.

Données d'entrée

3
5
7

Données de sortie

6

Justification des données de sortie

Le coût de chaque case est indiqué. Les cases qu'Ève doit acheter pour minimiser le coût de son chemin de jonction sont surlignées en vert.

Territoire du haut				
1	2	3	4	5
6	7	1	2	3
4	5	6	7	1
Territoire du bas				