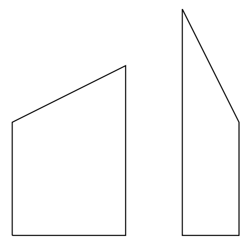
Problem S1: Crazy Fencing

Problem Description

You need to paint a wooden fence between your house and your neighbour's house. You want to determine the area of the fence, in order to determine how much paint you will use.

However, the fence is made out of N non-uniform pieces of wood, and your neighbour believes that they have an artistic flair. In particular, the pieces of wood may be of various widths. The bottom of each piece of wood will be horizontal, both sides will be vertical, but its top may be cut on an angle. Two such pieces of wood are shown below:



Thankfully, the fence has been constructed so that adjacent pieces of wood have the same height on the sides where they touch, which makes the fence more visually appealing.

Input Specification

The first line of the input will be a positive integer N, where $N \leq 10\,000$.

The second line of input will contain $N + 1$ space-separated integers $h_1, \ldots, h_{N+1}$ ($1 \leq h_i \leq 100$, $1 \leq i \leq N + 1$) describing the left and right heights of each piece of wood. Specifically, the left height of the i^{th} piece of wood is h_i and the right height of the i^{th} piece of wood is h_{i+1}.

The third line of input will contain N space-separated integers w_i ($1 \leq w_i \leq 100$, $1 \leq i \leq N$) describing the width of the i^{th} piece of wood.

Output Specification

Output the total area of the fence.

Sample Input 1

```
3
2 3 6 2
4 1 1
```

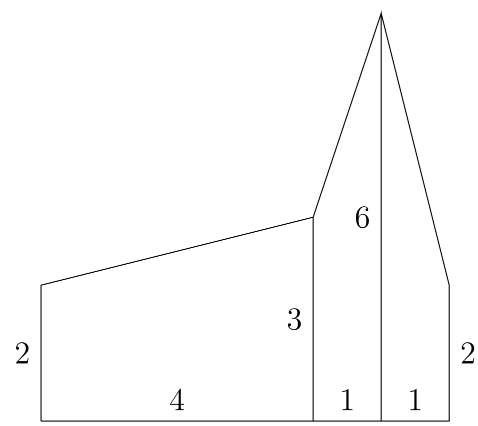
Output for Sample Input 1

```
18.5
```

Explanation of Output for Sample Input 1

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

The fence looks like the following:



When looking from left to right, the individual areas of the pieces of wood are $10 = 4{\cdot}(2{+}3)/2$, $4.5 = 1 \cdot (3+6)/2$, and $4 = 1 \cdot (6+2)/2$, for a total area of 18.5.

Sample Input 2

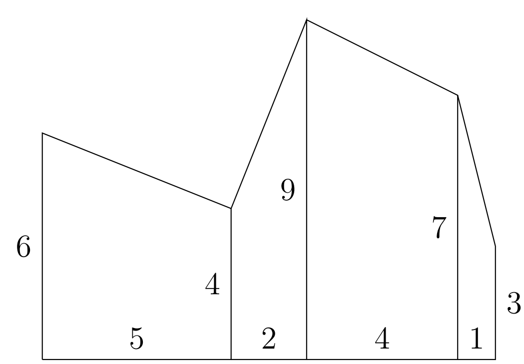
```
4
6 4 9 7 3
5 2 4 1
```

Output for Sample Input 2

```
75
```

Explanation of Output for Sample Input 2

The fence looks like the following:



When looking from left to right, the individual areas of the pieces of wood are 25, 13, 32, and 5, for a total area of 75.

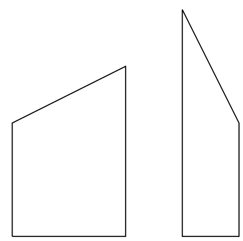
La version française figure à la suite de la version anglaise.

Problème S1 : Une drôle de clôture

Énoncé du problème

Vous devez peindre la clôture en bois séparant votre maison et celle du voisin. Vous souhaitez déterminer l'aire de la clôture afin de déterminer la quantité de peinture qu'il faudra acheter.

Cependant, la clôture est composée de N morceaux de bois non uniformes qui, selon votre voisin, lui donnent une touche artistique. En particulier, les morceaux de bois peuvent être de différentes largeurs. De plus, chaque morceau de bois a un côté inférieur horizontal et deux côtés parallèles verticaux. Or, le côté supérieur de chaque morceau de bois peut être coupé en angle. On voit deux tels morceaux de bois dans la figure ci-dessous :



Heureusement, la clôture a été construite de manière que les côtés où les morceaux de bois adjacents sont en contact sont de même hauteur, ce qui rend la clôture plus esthétique.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée ne contient qu'un seul entier strictement positif N tel que $N \leq 10\,000$.

La deuxième ligne des données d'entrée contient $N+1$ entiers $h_1, \ldots, h_{N+1}$ ($1 \leq h_i \leq 100$, $1 \leq i \leq N+1$) dont chacun est séparé des autres par un espace. Ces entiers représentent les hauteurs du côté gauche et du côté droit de chaque morceau de bois. Plus précisément, le côté gauche du i^{e} morceau de bois a une hauteur de h_i tandis que le côté droit du i^{e} morceau de bois a une hauteur de h_{i+1}.

La troisième ligne des données d'entrée contient N entiers w_i ($1 \leq w_i \leq 100$, $1 \leq i \leq N$) dont chacun est séparé des autres par un espace. Ces entiers représentent la largeur du i^{e} morceau de bois.

Précisions par rapport aux données de sortie

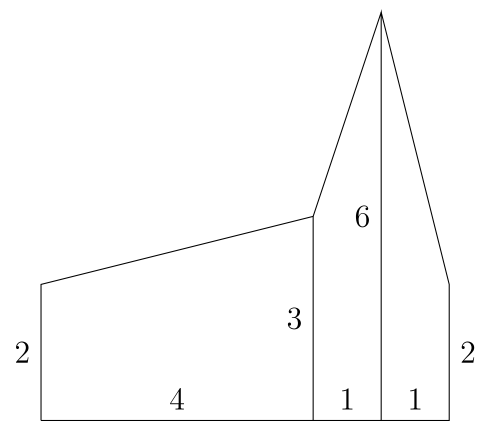
Les données de sortie ne devraient contenir que l'aire totale de la clôture.

Données d'entrée d'un 1^er^ exemple

```
3
2 3 6 2
4 1 1
```

Données de sortie du 1er exemple
```
18.5
```

Justification des données de sortie du 1er exemple
La clôture ressemble à la figure suivante :

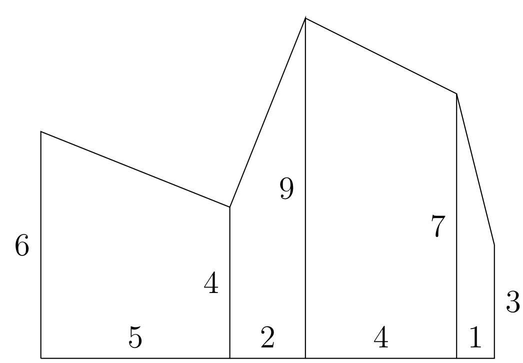


Allant de gauche à droite, les morceaux de bois ont les aires suivantes : $10 = 4 \cdot (2+3)/2$, $4,5 = 1 \cdot (3+6)/2$ et $4 = 1 \cdot (6+2)/2$, d'où l'aire totale de 18,5. (Remarquez que le système de notation a été programmé en anglais. Pour cette raison, veuillez utiliser le point comme signe décimal plutôt que la virgule dans les données de sortie.)

Données d'entrée d'un 2e exemple
```
4
6 4 9 7 3
5 2 4 1
```

Données de sortie du 2e exemple
```
75
```

Justification des données de sortie du 2e exemple
La clôture ressemble à la figure suivante :



English version appears before the French version

Allant de gauche à droite, les morceaux de bois ont les aires suivantes : 25, 13, 32 et 5, d'où l'aire totale de 75.

Problem J5/S2: Modern Art

Problem Description
A new and upcoming artist has a unique way to create checkered patterns. The idea is to use an M-by-N canvas which is initially entirely black. Then the artist repeatedly chooses a row or column and runs their magic brush along the row or column. The brush changes the colour of each cell in the row or column from black to gold or gold to black.

Given the artist's choices, your job is to determine how much gold appears in the pattern determined by these choices.

Input Specification
The first line of input will be a positive integer M. The second line of input will be a positive integer N. The third line of input will be a positive integer K. The remaining input will be K lines giving the choices made by the artist. Each of these lines will either be `R` followed by a single space and then an integer which is a row number, or `C` followed by a single space and then an integer which is a column number. Rows are numbered top down from 1 to M. Columns are numbered left to right from 1 to N.

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

1 mark	$M = 1$	$N = 1$	$K \leq 100$	only one cell, and up to 100 choices by the artist
4 marks	$M = 1$	$N \leq 100$	$K \leq 100$	only one row, and up to 100 choices by the artist
5 marks	$M \leq 100$	$N \leq 100$	$K \leq 100$	up to 100 rows, up to 100 columns, and up to 100 choices by the artist
5 marks	$MN \leq 5\,000\,000$		$K \leq 1\,000\,000$	up to 5 000 000 cells, and up to 1 000 000 choices by the artist

Output Specification
Output one non-negative integer which is equal to the number of cells that are gold in the pattern determined by the artist's choices.

Sample Input 1
```
3
3
2
R 1
C 1
```

Output for Sample Input 1
```
4
```

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

Explanation of Output for Sample Input 1

After running the brush along the first row, the canvas looks like this:

```
GGG
BBB
BBB
```

Then after running the brush along the first column, four cells are gold in the final pattern determined by the artist's choices:

```
BGG
GBB
GBB
```

Sample Input 2

```
4
5
7
R 3
C 1
C 2
R 2
R 2
C 1
R 4
```

Output for Sample Input 2

```
10
```

Explanation of Output for Sample Input 2

Ten cells are gold in the final pattern determined by the artist's choices:

```
BGBBB
BGBBB
GBGGG
GBGGG
```

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

Problème J5/S2 : L'art moderne

Énoncé du problème
Un nouvel artiste a développé une façon unique de créer des motifs en damier. L'artiste se procure d'abord une toile de couleur noire et de dimensions $M \times N$. Ensuite, l'artiste choisit à plusieurs reprises une rangée ou une colonne et donne un coup de son pinceau magique le long de la rangée ou de la colonne. Le pinceau change la couleur de chaque case de la rangée ou de la colonne du noir à l'or ou de l'or au noir.

Étant donné les choix de l'artiste, votre tâche consiste à déterminer le nombre de cases dorées qui paraissent dans le motif en damier résultant.

Précisions par rapport aux données d'entrée
La première ligne des données d'entrée ne contient qu'un seul entier strictement positif, soit M. La deuxième ligne des données d'entrée ne contient qu'un seul entier strictement positif, soit N. La troisième ligne des données d'entrée ne contient qu'un seul entier strictement positif, soit K. Le restant des données d'entrée sera composé de K lignes ; ces lignes représentant les choix de l'artiste. Chacune de ces K lignes commencera par `R` ou par `C` (indiquant respectivement une rangée ou une colonne) suivi d'un seul espace puis d'un entier strictement positif inférieur ou égal à N. Cet entier représente le numéro d'une rangée ou d'une colonne. Les rangées sont numérotées de haut en bas de 1 à M. Les colonnes sont numérotées de gauche à droite de 1 à N.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis.

1 point	$M = 1$	$N = 1$	$K \leq 100$	Une seule case, et usqu'à 100 choix de l'artiste
4 points	$M = 1$	$N \leq 100$	$K \leq 100$	Une seule rangée, et jusqu'à 100 choix de l'artiste
5 points	$M \leq 100$	$N \leq 100$	$K \leq 100$	Jusqu'à 100 rangées, et jusqu'à 100 colonnes, et jusqu'à 100 choix de l'artiste
5 points	$MN \leq 5\,000\,000$		$K \leq 1\,000\,000$	Jusqu'à 5 000 000 cases, et jusqu'à 1 000 000 choix de l'artiste

Précisions par rapport aux données de sortie
Les données de sortie ne devraient contenir qu'un seul entier non négatif. Cet entier est égal au nombre de cases dorées dans le motif en damier résultant des choix de l'artiste.

Données d'entrée d'un 1^er exemple

```
3
3
2
R 1
C 1
```

Données de sortie du 1^er exemple

```
4
```

Justification des données de sortie du 1^er exemple

Après avoir donné un coup de pinceau le long de la première rangée, la toile ressemble à ceci (`G` représente les cases dorées et `B` les cases noires) :

```
GGG
BBB
BBB
```

Ensuite, après avoir donné un coup de pinceau le long de la première colonne, il y a quatre cases dorées dans le motif en damier final de l'artiste :

```
BGG
GBB
GBB
```

Données d'entrée d'un 2^e exemple

```
4
5
7
R 3
C 1
C 2
R 2
R 2
C 1
R 4
```

Données de sortie du 2^e exemple

```
10
```

Justification des données de sortie du 2^e exemple

Il y a dix cases dorées dans le motif en damier final de l'artiste :

```
BGBBB
BGBBB
GBGGG
GBGGG
```

Problem S3: Lunch Concert

Problem Description

It's lunchtime at your school! Your N friends are all standing on a long field, as they usually do. The field can be represented as a number line, with the ith friend initially at position P_i metres along it. The ith friend is able to walk in either direction along the field at a rate of one metre per W_i seconds, and their hearing is good enough to be able to hear music up to and including D_i metres away from their position. Multiple students may occupy the same positions on the field, both initially and after walking.

You're going to hold a little concert at some position c metres along the field (where c is any integer of your choice), and text all of your friends about it. Once you do, each of them will walk along the field for the minimum amount of time such that they end up being able to hear your concert (in other words, such that each friend i ends up within D_i units of c).

You'd like to choose c such that you minimize the sum of all N of your friends' walking times. What is this minimum sum (in seconds)? Please note that the result might not fit within a 32-bit integer.

Input Specification

The first line of input contains N.
The next N lines contain three space-separated integers, P_i, W_i, and D_i ($1 \leq i \leq N$).

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

4 marks	$1 \leq N \leq 2000$	$0 \leq P_i \leq 2000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 2000$
9 marks	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000$
2 marks	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000\ 000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000\,000$

Output Specification

Output one integer which is the minimum possible sum of walking times (in seconds) for all N of your friends to be able to hear your concert.

Sample Input 1

```
1
0 1000 0
```

Output for Sample Input 1

```
0
```

Explanation of Output for Sample Input 1

If you choose $c = 0$, your single friend won't need to walk at all to hear it.

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

Sample Input 2

```
2
10 4 3
20 4 2
```

Output for Sample Input 2

```
20
```

Explanation of Output for Sample Input 2

One possible optimal choice of c is 14, which would require your first friend to walk to position 11 (taking $4 \times 1 = 4$ seconds) and your second friend to walk to position 16 (taking $4 \times 4 = 16$ seconds), for a total of 20 seconds.

Sample Input 3

```
3
6 8 3
1 4 1
14 5 2
```

Output for Sample Input 3

```
43
```

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

Problème S3 : Déjeuner-concert

Énoncé du problème
C'est l'heure du déjeuner à votre école ! Comme d'habitude, vos N amis sont tous regroupés sur un long terrain adjacent à votre école. Le terrain peut être représenté par une droite numérique où votre i^{e} ami est situé à une position intiale (soit P_i) en mètres le long de cette droite numérique. Votre i^{e} ami est capable de marcher dans les deux sens le long du terrain à une vitesse de 1 mètre par W_i secondes. De plus, son ouïe est suffisamment bonne pour pouvoir entendre de la musique jusqu'à D_i metres de sa position. Plusieurs élèves peuvent occuper les mêmes positions sur le terrain, à la fois avant et après avoir marché.

Vous organisez un petit concert à une certaine position (soit c) en mètres le long du terrain (c étant un entier de votre choix) et vous envoyez un SMS à tous vous amis pour les en informer. Une fois que vous l'avez fait, chacun d'eux marchera le long du terrain pendant le minimum de temps nécessaire pour pouvoir entendre votre concert (autrement dit, de sorte que chaque ami i soit situé à D_i unités ou moins de c).

Vous souhaitez choisir c de manière à minimiser la somme de tous les temps de marche de vos N amis. Quelle est cette somme minimale (en secondes) ? Veuillez noter que le résultat peut dépasser les paramètres d'un entier 32 bits.

Précisions par rapport aux données d'entrée
La première ligne des données d'entrée contient N.
Les N prochaines lignes contiennent trois entiers P_i, W_i et D_i ($1 \leq i \leq N$), dont chacun est séparé des autres par un espace.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis.

4 points	$1 \leq N \leq 2000$	$0 \leq P_i \leq 2000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 2000$
9 points	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000$
2 points	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000\;000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000\,000$

Précisions par rapport aux données de sortie
Les données de sortie ne devraient contenir qu'un seul entier. Ce dernier représente la somme minimale possible de tous les temps de marche (en secondes) de vos N amis afin qu'ils puissent tous entendre le concert.

Données d'entrée d'un 1^{er} exemple

```
1
0 1000 0
```

Données de sortie du 1^{er} exemple

```
0
```

Justification des données de sortie du 1^er^ exemple

Si vous choisissez $c = 0$, votre seul ami n'aura pas besoin de se déplacer pour pouvoir entendre le concert.

Données d'entrée d'un 2^e^ exemple

```
2
10 4 3
20 4 2
```

Données de sortie du 2^e^ exemple

```
20
```

Justification des données de sortie du 2^e^ exemple

On peut avoir 14 comme choix optimal possible de c. Ceci obligerait votre premier ami à marcher jusqu'à la position 11 (en $4 \times 1 = 4$ secondes) et votre second ami à marcher jusqu'à la position 16 (en $4 \times 4 = 16$ secondes), soit un total de 20 secondes.

Données d'entrée d'un 3^e^ exemple

```
3
6 8 3
1 4 1
14 5 2
```

Données de sortie du 3^e^ exemple

```
43
```

Problem S4: Daily Commute

Problem Description

Toronto has N subway stations, numbered from 1 to N. You start at station 1, and every day, you need to reach station N to get to school.

There are W *one-way* walkways running amongst the stations, the i^{th} of which allows you to walk *from* station A_i *to* a different station B_i ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$) in 1 minute. There may be multiple walkways connecting any given pair of stations.

The subway line follows a certain route through the N stations, starting at station 1 and visiting each station once. Initially, this route consists of stations $S_1, S_2, ..., S_N$, in that order. $S_1 = 1$, and $S_2, \ldots, S_N$ is a permutation of the integers $2, \ldots, N$. Only one subway train runs along this route per day, departing from station 1 at 6am in the morning and taking 1 minute to reach each subsequent station. This means that, m minutes after 6am, the train will be at station S_{m+1} (or at station S_N if $m \geq N - 1$).

Over a period of D days, however, the subway line's route will keep changing. At the start of the i^{th} day, the ${X_i}^{\text{th}}$ station and ${Y_i}^{\text{th}}$ station ($2 \leq X_i, Y_i \leq N$, $X_i \neq Y_i$) in the route will be swapped. Note that, after each such change, the route will still begin at station 1 and will visit all N stations once each. Changes will carry over to subsequent days – the route will not automatically reset itself back to $S_1, \ldots, S_N$.

On each of these D days, you'd like to determine how quickly you can get to school so you can begin learning things. On the i^{th} day, starting at 6am in the morning (after the i^{th} update to the subway line's route), you'll begin your daily trip to station N. Each minute, you may either ride the subway to its next stop (if you're currently at the same station as the train and it hasn't already completed its route), take a walkway from your current station to another one, or wait at your current station. Note that your trip begins at the same time as the train's route, meaning that you may choose to immediately ride it if you'd like to, and that you may choose to leave and then get back on the train during your trip.

Input Specification

The first line contains three space-separated integers, N, W, and D.
The next W lines each contain two space-separated integers, A_i and B_i ($1 \leq i \leq W$).
The next line contains the N space-separated integers, $S_1, \ldots, S_N$, which form the initial permutation of stations.
The next D lines each contain two space-separated integers, X_i and Y_i ($1 \leq i \leq D$).

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

2 marks	$3 \leq N \leq 10$	$0 \leq W \leq 10$	$1 \leq D \leq 10$
2 marks	$3 \leq N \leq 200$	$0 \leq W \leq 200$	$1 \leq D \leq 200$
3 marks	$3 \leq N \leq 2000$	$0 \leq W \leq 2000$	$1 \leq D \leq 2000$
8 marks	$3 \leq N \leq 200\,000$	$0 \leq W \leq 200\,000$	$1 \leq D \leq 200\,000$

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

Output Specification

The output is D lines, with one integer per line. The i^{th} line is the minimum number of minutes required to reach station N on the i^{th} day ($1 \leq i \leq D$).

Sample Input

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

Output for Sample Input

```
1
2
3
```

Explanation of Output for Sample Input

At the start of the first day, the subway line's route will be updated to visit stations $[1, 4, 2, 3]$, in that order. On that day, you should simply take the subway to station 4, taking 1 minute.

On the second day, the route will become $[1, 3, 2, 4]$, and you should take the subway to station 3 (taking 1 minute) and then walk to station 4 (taking 1 more minute).

On the third day, the route will become $[1, 2, 3, 4]$. One choice of optimal trip involves walking to station 2 (taking 1 minute), then boarding the train there and taking it through station 3 and finally to station 4 (taking another 2 minutes).

Problème S4 : Trajet quotidien

Énoncé du problème

Toronto a N stations de métro. Ces dernières sont numérotées de 1 à N. Vous commencez à la station 1 et vous devez vous rendre à la station N pour aller à l'école chaque jour.

Il y a W passerelles *à sens unique* entre les stations. La i^{e} passerelle vous permet de marcher *de* la station A_i *à* une station différente B_i ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$) en 1 minute. Il peut y avoir plusieurs passerelles reliant un couple de stations donné.

La ligne de métro suit un certain itinéraire à travers les N stations en commençant à la station 1 et en passant par chaque station une seule fois. Selon l'itinéraire initial, le train passait par les stations dans l'ordre suivant : $S_1, S_2, ..., S_N$ où $S_1 = 1$ tandis que $S_2, \ldots, S_N$ est une permutation des entiers $2, \ldots, N$. Un seul train de métro circule le long de cet itinéraire par jour, partant de la station 1 à 6 heures du matin et prenant 1 minute pour atteindre chaque station suivante. Cela signifie que le train arrivera à la station S_{m+1} à m minutes après 6 h (ou à la station S_N si $m \geq N - 1$).

Cependant, l'itinéraire de la ligne de métro continuera à changer sur une période de D jours. Au début du i^{e} jour, on échange les positions de la $X_i{}^{\text{e}}$ station et la $Y_i{}^{\text{e}}$ station ($2 \leq X_i, Y_i \leq N$, $X_i \neq Y_i$) dans l'itinéraire. Remarquez qu'après chacun de ces changements, l'itinéraire commencera toujours à la station 1 et ne passera par chacune des N stations qu'une seule fois. De plus, tous changements qui sont apportés à l'itinéraire seront reportés aux jours suivants – l'itinéraire ne se réinitialisera pas automatiquement à $S_1, \ldots, S_N$.

À chacun de ces D jours, vous souhaitez déterminer la manière la plus rapide de vous rendre à l'école pour rassasier votre soif de connaissance. Vous commencez votre trajet quotidien jusqu'à la station N au i^{e} jour à 6 h du matin (après la i^{e} mise à jour de l'itinéraire de la ligne de métro). Chaque minute, vous pouvez soit prendre le métro jusqu'à son prochain arrêt (si vous êtes actuellement dans la même station que le train et qu'il n'a pas déjà terminé son itinéraire), soit emprunter une passerelle de votre station actuelle à une autre, soit attendre à votre station actuelle. Remarquez que votre voyage commence en même temps que l'itinéraire du train. Cela signifie que vous pouvez prendre le train immédiatement si vous le souhaitez mais que vous pouvez aussi quitter le train et le reprendre plus tard dans le trajet.

Précisions par rapport aux données d'entrée

La première ligne contient trois entiers, soit N, W et D, dont chacun est séparé des autres par un espace.
Chacune des W prochaines lignes contient deux entiers, soit A_i et B_i ($1 \leq i \leq W$), les deux étant séparés par un espace.
La prochaine ligne contient N entiers, soit $S_1, \ldots, S_N$, dont chacun est séparé des autres par un espace. Ces entiers représentent l'ordre des stations dans l'itinéraire initial.
Chacune des D prochaines lignes contient deux entiers, soit X_i et Y_i ($1 \leq i \leq D$), les deux étant séparés par un espace.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis.

2 points	$3 \leq N \leq 10$	$0 \leq W \leq 10$	$1 \leq D \leq 10$
2 points	$3 \leq N \leq 200$	$0 \leq W \leq 200$	$1 \leq D \leq 200$
3 points	$3 \leq N \leq 2000$	$0 \leq W \leq 2000$	$1 \leq D \leq 2000$
8 points	$3 \leq N \leq 200\,000$	$0 \leq W \leq 200\,000$	$1 \leq D \leq 200\,000$

Précisions par rapport aux données de sortie

Les données de sortie devraient contenir D lignes. Chacune des lignes ne devrait contenir qu'un seul entier. La i^{e} ligne est le nombre minimal de minutes nécessaires pour atteindre la station N au i^{e} jour ($1 \leq i \leq D$).

Exemple de données d'entrée

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

Exemple de données de sortie

```
1
2
3
```

Justification des données de sortie

Au début du premier jour, l'itinéraire de la ligne de métro subit une mise à jour. Selon le nouvel itinéraire, le train passera par les stations dans l'ordre suivant : $[1, 4, 2, 3]$. Ce jour-là, il suffit simplement de prendre le train jusqu'à la station 4. Ce trajet durera 1 minute.

Au deuxième jour, le nouvel itinéraire est $[1, 3, 2, 4]$. Dans ce cas, vous devrez prendre le train jusqu'à la station 3 (ce qui vous prendra 1 minute) et ensuite marcher jusqu'à la station 4 (ce qui vous prendra 1 minute additionnelle).

Au troisième jour, le nouvel itinéraire est $[1, 2, 3, 4]$. Un trajet optimal possible consiste à marcher jusqu'à la station 2 (ce qui vous prendra 1 minute), puis à prendre le train à cette station pour ensuite descendre à la station 4 (ce qui vous prendra 2 minutes additionnelles).

Problem S5: Math Homework

Problem Description

Your math teacher has given you an assignment involving coming up with a sequence of N integers $A_1, \ldots, A_N$, such that $1 \leq A_i \leq 1\ 000\ 000\ 000$ for each i.

The sequence A must also satisfy M requirements, with the i^{th} one stating that the GCD (Greatest Common Divisor) of the contiguous subsequence $A_{X_i}, \ldots, A_{Y_i}$ $(1 \leq X_i \leq Y_i \leq N)$ must be equal to Z_i. Note that the GCD of a sequence of integers is the largest integer d such that all the numbers in the sequence are divisible by d.

Find *any* valid sequence A consistent with all of these requirements, or determine that no such sequence exists.

Input Specification

The first line contains two space-separated integers, N and M.
The next M lines each contain three space-separated integers, X_i, Y_i, and Z_i $(1 \leq i \leq M)$.

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

3 marks	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 2$ for each i
4 marks	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 16$ for each i
8 marks	$1 \leq N \leq 150\,000$	$1 \leq M \leq 150\,000$	$1 \leq Z_i \leq 16$ for each i

Output Specification

If no such sequence exists, output the string `Impossible` on one line. Otherwise, on one line, output N space-separated integers, forming the sequence $A_1, \ldots, A_N$. If there are multiple possible valid sequences, *any* valid sequence will be accepted.

Sample Input 1

```
2 2
1 2 2
2 2 6
```

Output for Sample Input 1

```
4 6
```

Explanation of Output for Sample Input 1

If $A_1 = 4$ and $A_2 = 6$, the GCD of $[A_1, A_2]$ is 2 and the GCD of $[A_2]$ is 6, as required. **Please note that other outputs would also be accepted.**

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

Sample Input 2

```
2 2
1 2 2
2 2 5
```

Output for Sample Input 2

```
Impossible
```

Explanation of Output for Sample Input 2

There exists no sequence $[A_1, A_2]$ such that the GCD of $[A_1, A_2]$ is 2 and the GCD of $[A_2]$ is 5.

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

Problème S5 : Devoirs de maths

Énoncé du problème

Votre professeur de mathématiques vous a donné un travail consistant à créer une suite de N entiers $A_1, \ldots, A_N$ telle que $1 \leq A_i \leq 1\ 000\ 000\ 000$ pour chaque i.

La suite A doit également répondre à M critères, dont le i^e qui stipule que le PGCD (le plus grand commun diviseur) de la sous-suite contiguë (ou sous-séquence contiguë) $A_{X_i}, \ldots, A_{Y_i}$ $(1 \leq X_i \leq Y_i \leq N)$ doit être égal à Z_i. Remarquez que le PGCD d'une suite d'entiers est le plus grand entier d tel que tous les nombres de la suite soient divisibles par d.

Trouvez *une* suite valide A qui répond à tous ces critères, ou déterminez qu'une telle suite n'existe pas.

Précisions par rapport aux données d'entrée

La première ligne des données d'entrée contient deux entiers, soit N et M, les deux étant séparés par un espace.
Chacune des M prochaines lignes contient trois entiers, soit X_i, Y_i et Z_i $(1 \leq i \leq M)$, dont chacun est séparé des autres par un espace.

Le tableau suivant indique la manière dont les 15 points disponibles sont répartis.

3 points	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 2$ pour chaque i
4 points	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 16$ pour chaque i
8 points	$1 \leq N \leq 150\,000$	$1 \leq M \leq 150\,000$	$1 \leq Z_i \leq 16$ pour chaque i

Précisions par rapport aux données de sortie

Si une telle suite n'existe pas, les données de sortie devraient afficher `Impossible` sur une seule ligne. Sinon, les données de sortie ne devraient contenir qu'une seule ligne ; cette dernière contiendra N entiers qui seront séparés les uns des autres par un espace afin de former la suite $A_1, \ldots, A_N$. S'il existe plusieurs suites valides possibles, *n'importe laquelle* des suites valides sera acceptée.

Données d'entrée d'un 1[er] exemple

```
2 2
1 2 2
2 2 6
```

Données de sortie du 1[er] exemple

```
4 6
```

Justification des données de sortie du 1[er] exemple

Si $A_1 = 4$ et $A_2 = 6$, le PGCD de $[A_1, A_2]$ est 2 et le PGCD de $[A_2]$ est 6, ce qu'il fallait démontrer. **Veuillez noter que d'autres suites seraient également acceptées.**

Données d'entrée d'un 2e exemple

```
2 2
1 2 2
2 2 5
```

Données de sortie du 2e exemple

```
Impossible
```

Justification des données de sortie du 2e exemple

Il n'existe aucune suite $[A_1, A_2]$ telle que le PGCD de $[A_1, A_2]$ soit 2 et que le PGCD de $[A_2]$ soit 5.