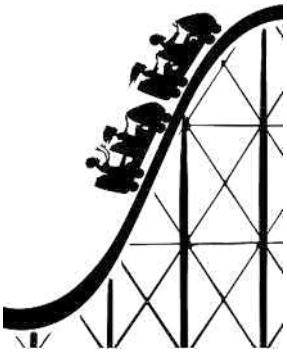
Problem J1: Roller Coaster Ride



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.

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 $10 + 24 - 6 - 12 = 16$.

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
```

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.

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
```

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.

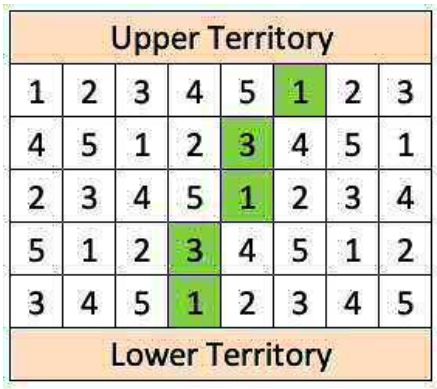
Problem J5: Connecting Territories

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: $1, 2, 3, \ldots, M, 1, 2, 3, \ldots, M, 1, 2, 3, \ldots$. 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.



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.

Sample Input

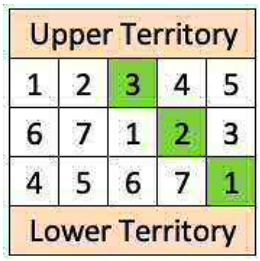
3
5
7

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.



Problem S1: Positioning Peter's Paintings

Problem Description
Peter the painter just finished painting two rectangular paintings and would like to display both on a rectangular wall which has the smallest perimeter possible. The first painting has a base of length A units and a height of length B units. The second painting has a base of length X units and a height of length Y units.

Peter has a few conditions on how to arrange his paintings on the rectangular wall. The first condition is that the paintings must be upright, meaning that the bases of the paintings are parallel to the floor. The second condition is that he would like to display both paintings in full, meaning that they cannot overlap each other. Please help determine the rectangular wall of minimum perimeter such that the paintings can be displayed without violating his conditions.

Input Specification
The one line of input will consist of four space-separated positive integers, A, B, X, Y ($1 \leq A, B, X, Y \leq 10^8$).

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

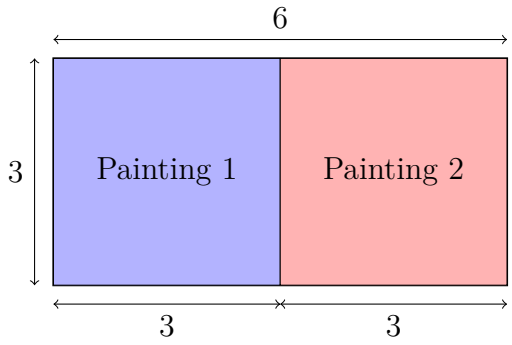
Marks	Brief Description
5	Paintings are congruent squares
5	Paintings are squares
5	Paintings are rectangles (possibly squares)

Output Specification
Output a single integer representing the minimum perimeter of a rectangular wall without violating Peter's conditions.

Sample Input 1
```
3 3 3 3
```

Output for Sample Input 1
```
18
```

Explanation of Output for Sample Input 1
This test case satisfies all subtasks. An optimal arrangement using a 6-by-3 wall is shown below.



Sample Input 2

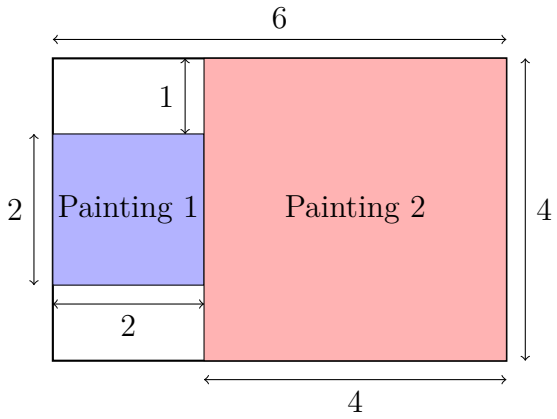
```
2 2 4 4
```

Output for Sample Input 2

```
20
```

Explanation of Output for Sample Input 2

This test case satisfies the second and third subtasks. An optimal arrangement using a 6-by-4 wall is shown below.



Sample Input 3

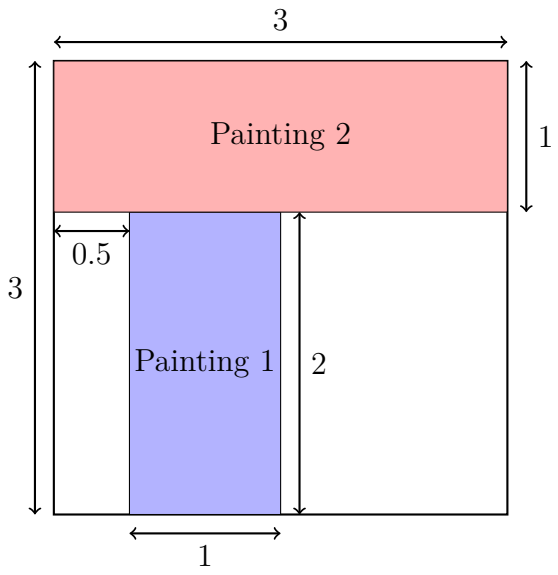
```
1 2 3 1
```

Output for Sample Input 3

```
12
```

Explanation of Output for Sample Input 3

This test case satisfies the last subtask. An optimal arrangement using a 3-by-3 wall is shown below.


3
Painting 2
1
0.5
3
Painting 1
2
1

Problem S2: Cryptogram Cracking Club

Problem Description
Cyrene, the captain of the Cryptogram Cracking Club (CCC), came across a concerningly long cipher. Conveniently, this cipher is composed of lower-case characters (`a-z`). Comfortingly, the cipher is composed of a pattern that repeats infinitely.

Cyrene wishes to locate the c-th character of the cipher. To make your job easier, the CCC members have extracted the repeated pattern and compressed it using the Run-Length Encoding (RLE) algorithm, which replaces consecutive repeated characters with a single occurrence of the character followed by a count of how many times it was repeated. For example, for the pattern `aaaabccdddd`, the RLE algorithm outputs `a4b1c2d4`.

You are given the output of the RLE algorithm for a certain pattern. Can you determine the c-th character of the long cipher that is formed by repeating this pattern infinitely?

Input Specification
The first line of input will consist of a string S, representing a pattern produced by the RLE algorithm. The length of S will be at least 2 and at most $2 \cdot 10^5$. Additionally, all numbers appearing in S are between 1 and 10^{12}.

The next line of input contains a single integer c, representing the index of the character you wish to locate, starting from index 0.

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

Marks	Bounds on c	Additional Constraints
6	$0 \leq c \leq 2000$	All numbers appearing in S are between 1 and 9 (inclusive) and the length of the repeated pattern is at most 2000 characters.
3	$0 \leq c \leq 10^6$	The length of the repeated pattern is at most 10^6 characters.
3	$0 \leq c \leq 10^{12}$	The length of the repeated pattern is at most 10^6 characters.
3	$0 \leq c \leq 10^{12}$	No additional constraints.

Output Specification
Output the c-th character of the long cipher.

Sample Input 1
```
r2d2
8
```

Output for Sample Input 1

```
r
```

Explanation of Output for Sample Input 1

The output of the RLE algorithm `r2d2` corresponds to the pattern `rrdd`, which creates the infinitely long cipher `rrddrrdd`r`rddrrdd...`, where the $c = 8$th character is `r`. In this example, the $c = 8$th character is highlighted with a box around it.

Sample Input 2

```
a4b1c2d10
100
```

Output for Sample Input 2

```
d
```

Explanation of Output for Sample Input 2

The output of the RLE algorithm `a4b1c2d10` corresponds to the pattern `aaaabccddddddddddd`. When repeated infinitely, the $c = 100$th character is `d`.

Problem S3: Pretty Pens

Problem Description
You are taking an art class, and your current art assignment is very algorithmic.

You have N pens, each of which has a single colour, represented by an integer from 1 to M. Initially, the colour of the i-th pen is given by C_i. In addition, your pens are pretty, with the i-th pen having a prettiness of P_i.

For your assignment, you need to create a picture using M strokes, each from a pen of a different colour. The prettiness of your picture is the sum of the prettiness of the pens used to create the picture.

Your teacher has given you some room for artistic expression: before you create this pretty picture, you are allowed to change at most one pen to any other colour. After this picture is drawn, the pen will revert back to its original colour.

Your teacher will give you $\frac{1}{3}$ of the marks (5 of 15 total marks) for the art assignment based on creating the prettiest picture you can.

To push your creative limits, your teacher also has Q more pictures for you to create, which compose the remaining $\frac{2}{3}$ of the marks (10 of 15 total marks) for your art assignment. Before each picture, there will be one of two possible changes to the set of pens available:

- `1 i c` indicates that the colour of the i-th pen changes to c.
- `2 i p` indicates that the prettiness of the i-th pen changes to p.

The changes are executed sequentially (so the first change modifies the initial setup, the second change modifies the result of applying the first change, and so on).

As in the first picture you created, you are allowed to change the colour of at most one pen before the picture is created. Note that if you do change the colour of one pen, it affects only the next picture you draw, and the pen will revert to its previous colour before the next change is applied (if any).

What is the prettiness of the prettiest $Q + 1$ pictures you can create?

Input Specification
The first line of input contains three space-separated integers N, M, and Q ($1 \leq M \leq N \leq 200\,000$, $0 \leq Q \leq 200\,000$).

The next N lines each contain two space-separated integers, C_i and P_i ($1 \leq C_i \leq M, 1 \leq P_i \leq 10^9$), indicating the colour and prettiness of the i-th pen.

The next Q lines each contain three space-separated integers, beginning with 1 or 2. If the first integer is 1, it is followed by two integers i_j and c_j ($1 \leq i_j \leq N, 1 \leq c_j \leq M$),

representing a change of the colour of the i_j-th pen to c_j. If the first integer is 2, it is followed by two integers i_j and p_j ($1 \leq i_j \leq N, 1 \leq p_j \leq 10^9$), representing a change of the prettiness of the i_j-th pen to p_j.

It is guaranteed that initially and after each change, there is at least one pen with each of the M colours.

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

Marks	**Bounds on N and M**	**Bounds on Q**	**Additional constraints**
5	$1 \leq M \leq N \leq 200\,000$	$Q = 0$	None
2	$M = 1$; $1 \leq N \leq 200\,000$	$0 \leq Q \leq 200\,000$	None
2	$M = 2$; $2 \leq N \leq 200\,000$	$0 \leq Q \leq 200\,000$	None
2	$M \leq 10$; $1 \leq M \leq N \leq 200\,000$	$0 \leq Q \leq 200\,000$	None
2	$1 \leq M \leq N \leq 200\,000$	$0 \leq Q \leq 200\,000$	All prettiness values are distinct
2	$1 \leq M \leq N \leq 200\,000$	$0 \leq Q \leq 200\,000$	None

Output Specification
The output is $Q + 1$ lines, with the j-th line containing one integer, the largest possible prettiness value obtainable after the first $j - 1$ changes have been performed.

Sample Input 1
```
6 3 0
1 6
2 9
3 4
2 7
3 9
1 3
```

Output for Sample Input 1
```
25
```

Explanation of Output for Sample Input 1
There are six pens available in three different colours. The set of pens is:

- two pens of colour 1, with prettiness 6 and 3,
- two pens of colour 2, with prettiness 9 and 7,
- two pens of colour 3, with prettiness 4 and 9.

If we do not change the colour of any pens, the prettiest picture has prettiness $6+9+9=24$. However, if we change the colour of pen 4 from colour 2 to colour 1, we can create a picture with prettiness $7+9+9=25$, which is the prettiest picture that can be created.

Sample Input 2

```
3 2 2
1 20
2 30
1 10
1 3 2
2 3 25
```

Output for Sample Input 2

```
50
50
55
```

Explanation of Output for Sample Input 2

There are three pens with two different colours available.

Before the first change, using pen 1 and pen 2, with colours 1 and 2, respectively, achieves a prettiness of $20+30=50$.

After the first change, pen 3 has colour 2. There is no alteration in the maximum prettiness, even if we could switch one pen: picking pens 1 and 2 will yield the maximum prettiness.

After the second change, pen 3 will have colour 2 and prettiness 25. Before the final picture is created, we can change the colour of pen 2 to colour 1, and use pens 2 and 3 to achieve the maximum prettiness of $30+25=55$.

Problem S4: Floor is Lava

Problem Description

You're trapped in a scorching dungeon with N rooms numbered from 1 to N connected by M tunnels. The i-th tunnel connects rooms a_i and b_i in both directions, but the floor of the tunnel is covered in lava with temperature c_i.

To help you navigate the lavatic tunnels, you are wearing a pair of heat-resistant boots that initially have a chilling level of 0. In order to step through lava with temperature ℓ, your boots must have the same chilling level ℓ; if the chilling level is too low then the lava will melt your boots, and if it's too high then your feet will freeze as you cross the tunnel.

Luckily, when you're standing in a room, you can increase or decrease the chilling level of your boots by d for a cost of d coins. You start in room 1 and would like to reach the exit which you know is located in room N. What is the minimum cost to do so?

Input Specification

The first line of input contains two integers N and M ($1 \leq N, M \leq 200\,000$).

The next M lines each contain three integers a_i, b_i, and c_i ($1 \leq a_i, b_i \leq N, a_i \neq b_i, 1 \leq c_i \leq 10^9$), describing the i-th tunnel.

There is at most one tunnel connecting any pair of rooms, and it is possible to reach all other rooms from room 1.

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

Marks	Additional constraints
2	$M = N - 1$
4	For all tunnels, $1 \leq c_i \leq 10$
4	Each room has at most 5 outgoing tunnels
5	None

Output Specification

Output the minimum cost (in coins) to reach room N from room 1.

Sample Input

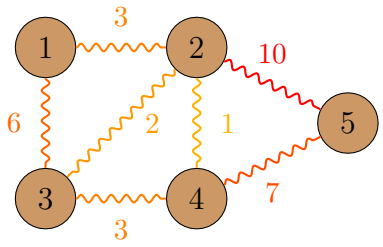
```
5 7
1 2 3
2 3 2
1 3 6
3 4 3
4 5 7
```

```
2 4 1
2 5 10
```

Output for Sample Input

```
9
```

Explanation of Output for Sample Input



A diagram of the dungeon is shown above. The optimal escape strategy is as follows:

1. Increase the chilling level to 3 for a cost of 3 coins.
2. Walk through the tunnel to room 2.
3. Decrease the chilling level to 2 for a cost of 1 coin.
4. Walk through the tunnel to room 3.
5. Increase the chilling level to 3 for a cost of 1 coin.
6. Walk through the tunnel to room 4.
7. Increase the chilling level to 7 for a cost of 4 coins.
8. Walk through the tunnel to room 5 and escape.

This has a total cost of 9 coins, and it can be shown that no cheaper route exists.

Problem S5: To-Do List

Problem Description
Wow, your to-do list is empty...but not for long! Over the next few seconds, you'll have to handle Q updates to your to-do list.

For the first type of update, you will have to add a new homework assignment to your to-do list. This assignment will be released at the beginning of second s, and will take t seconds to complete ($1 \leq s, t \leq 10^6$). For the second type of update, you will have to remove the i-th homework assignment that was added to your to-do list.

After each update, you wonder: what's the earliest time you can finish all of the homework assignments in your to-do list? You can only work on one assignment at a time, and you must finish a homework assignment once you start it without switching to another assignment.

Input Specification
The first line of input contains an integer Q.

The next Q lines each contain a line starting with a character `A` or `D`. A line starting with `A` represents the first type of update and ends with two space-separated **encrypted*** integers s' and t'. A line starting with `D` represents the second type of update and ends with an encrypted integer i'. It is guaranteed that there have been at least i assignments added and that the i-th assignment to be added has not been removed yet.

It is guaranteed that there is at least one homework assignment on your to-do list after every update.

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

Marks	Bounds on Q	Additional constraints
2	$1 \leq Q \leq 3\,000$	None
6	$1 \leq Q \leq 10^6$	Only updates of the first type
7	$1 \leq Q \leq 10^6$	None

*Note that the input for this problem is encrypted. To decrypt and obtain the actual values of s, t, and i, you may use the following formulas:

$$s = (s' + \text{ans}) \bmod (10^6 + 3) \quad t = (t' + \text{ans}) \bmod (10^6 + 3) \quad i = (i' + \text{ans}) \bmod (10^6 + 3)$$

Here, ans represents the answer after the previous update and is initially 0 before any updates. It may also be useful to note that mod corresponds to the `%` operator in most programming languages, indicating the remainder after division. For example, $5 \bmod 3 = 2$ and $17 \bmod 4 = 1$.

Output Specification

Output Q lines, where the i-th line contains the earliest time (in seconds) you can finish all of the homework assignments in your to-do list after the i-th update.

Sample Input 1

```
6
A 3 3
A 2 0
A 999996 999995
D 999991
A 1000000 999994
D 999992
```

Sample Input 1 (Unencrypted)

```
6
A 3 3
A 7 5
A 4 3
D 1
A 8 2
D 2
```

Output for Sample Input 1

```
5
11
13
11
13
9
```

Explanation of Output for Sample Input 1

The unencrypted sample input is provided for ease of reference.

After the first update, we can start the first assignment at the beginning of second 3 and finish at the end of second 5 (interval $[3, 5]$).

After the second update, we can do the first assignment over the interval $[3, 5]$ and the second assignment over the interval $[7, 11]$.

After the third update, we can do the first assignment over the interval $[3, 5]$, the third assignment over the interval $[6, 8]$, and then the second assignment over the interval $[9, 13]$.

After the fourth update, we can do the third assignment over the interval $[4, 6]$ and the second assignment over the interval $[7, 11]$.

After the fifth update, we can do the third assignment over the interval $[4, 6]$, the second

assignment over the interval $[7, 11]$, and the fourth assignment over the interval $[12, 13]$.

After the sixth update, we can do the third assignment over the interval $[4, 6]$ and the fourth assignment over the interval $[8, 9]$.

Sample Input 2

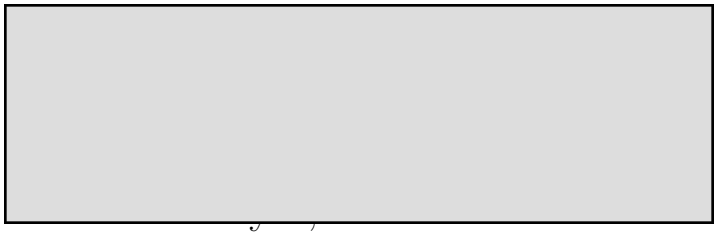
```
2
A 1000000 1000000
A 4 4
```

Sample Input 2 (Unencrypted)

```
2
A 1000000 1000000
A 1000000 1000000
```

Output for Sample Input 2

```
1999999
2999999
```



Asteroid Mining

Time Limit: 3 seconds

Problem Description
It is the year 2217 and Ryan is an asteroid miner. He makes a living by mining asteroids and selling them at the CCO (Celestial Cargo Outpost).

On his latest mining expedition, he has mined N mineral chunks where the i-th chunk has a value v_i and a mass m_i. Ryan plans to transport a set of chunks to the CCO with his rocket, but he only has enough fuel to last one more trip. He calculated that the maximum total mass he can safely carry on his rocket is M. Due to Ryan's mining technique, the chunks exhibit a special property: for any two mineral chunks, one's mass is divisible by the other chunk's mass.

Help Ryan find the maximum total value he can ship to CCO while adhering to his rocket's constraints.

Input Specification
The first line will contain two space-separated integers N ($1 \leq N \leq 500\,000$) and M ($1 \leq M \leq 10^{12}$).

The next N lines will each contain two space-separated integers v_i ($1 \leq v_i \leq 10^{12}$) and m_i ($1 \leq m_i \leq 10^{12}$), representing the value and mass of the i-th mineral chunk respectively. **Additionally, for any two mineral chunks i, j ($1 \leq i, j \leq N$), either $m_i \mid m_j$ or $m_j \mid m_i$, where $a \mid b$ means that a is a divisor of b (i.e., b/a is an integer).**

The following table shows how the available 25 marks are distributed:

Marks Awarded	Bounds on N	Bounds on M	Additional Constraints
2 marks	$N = 2$	$1 \leq M \leq 10^4$	None
2 marks	$1 \leq N \leq 20$	$1 \leq M \leq 10^4$	None
4 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 10^4$	None
6 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 10^8$	None
2 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^8$	All m_i are equal.
3 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^8$	At most 2 distinct m_i.
6 marks	$1 \leq N \leq 500\,000$	$1 \leq M \leq 10^{12}$	None

Output Specification
On one line, output one integer, the maximum total value Ryan can ship to CCO.

Sample Input
```
6 10
1 1
5 2
200 6
9 2
6 2
100 1
```

Output for Sample Input
```
310
```

Explanation of Output for Sample Input
Ryan can take all the chucks except the second and fifth chucks to achieve a total value of $1 + 200 + 9 + 100 = 310$. Note that the total mass of the chunks is $1 + 6 + 2 + 1 = 10$. We can show that this is optimal.

Tree Decorations

Time Limit: 2 seconds

Problem Description
Mateo recently found the perfect decorations for his Christmas tree — more trees!

Specifically, his Christmas tree is a rooted tree T initially with M nodes, all painted green. He has another rooted tree D that he uses as a reference for his decorations. Mateo uses the following process to put on all of his decorations:

- For each node i in D, he creates a **copy** of the subtree rooted at i. Let this copy be C_i. Then, he paints the nodes of C_i red. Finally, he chooses some green node in T to be the parent of the root of C_i by connecting them with an edge.

After applying all the decorations, T ends up containing N nodes. Unfortunately, he realized that he had forgotten to record what D is! To make things worse, he accidentally spilled water on T, washing off all the colour from the nodes. After all that, he labels the root of T as 1, and then labels the rest of the nodes from 2 to N.

The only information he currently has is the final state of T, as well as M. Help him find the number of possible D that he could have started with, where two possibilities are considered different if they are structurally distinct.

Rooted trees A and B are said to be structurally identical if and only if they have the same number of nodes S, and there is a way to label A's nodes from 1 to S and B's nodes from 1 to S such that:

- Their roots are labeled the same.
- Nodes labeled x and y in A are connected by an edge if and only if nodes labeled x and y in B are connected by an edge.

Otherwise, A and B are considered structurally distinct.

Input Specification
The first line of input contains two space-separated integers N and M.

The next $N-1$ lines each contain two space-separated integers u_i and v_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i$), describing an edge in T connecting nodes u_i and v_i. **Note that T is rooted at node 1**.

The following table shows how the available 25 marks are distributed:

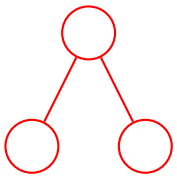
Marks Awarded	**Bounds on N**	**Bounds on M**
2 marks	$2 \leq N \leq 10$	$M = 1$
3 marks	$2 \leq N \leq 200$	$M = 1$
2 marks	$2 \leq N \leq 500\,000$	$M = 1$
6 marks	$2 \leq N \leq 200$	$1 \leq M < N$
4 marks	$2 \leq N \leq 2\,000$	$1 \leq M < N$
8 marks	$2 \leq N \leq 500\,000$	$1 \leq M < N$

Output Specification
Output the number of possible D that he could have started with, where two possibilities are considered different if they are structurally distinct.

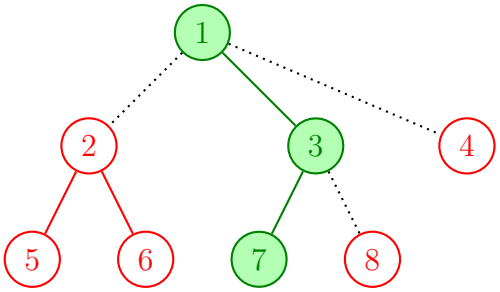
Sample Input 1
```
8 3
1 2
1 3
1 4
2 5
2 6
3 7
3 8
```

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

Explanation of Output for Sample Input 1
It is provable that the only possible D is:



We can get T the following way:



In the diagram above, the red parts are added as decorations, while the green, filled-in part represents the initial state of T. The dotted lines represent the edges connecting the decorations to the tree.

Sample Input 2

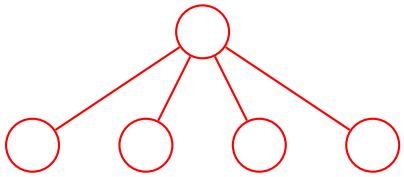
```
14 5
1 2
1 3
3 4
3 5
1 6
6 7
7 8
7 9
2 10
10 11
10 12
10 13
10 14
```

Output for Sample Input 2

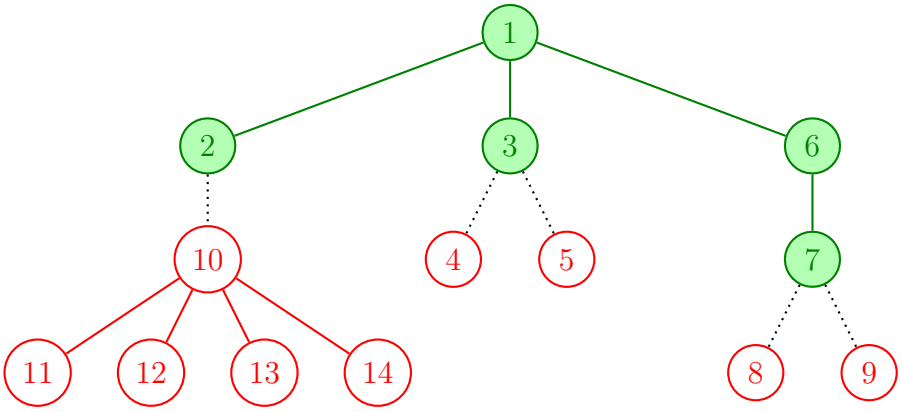
```
2
```

Explanation of Output for Sample Input 2

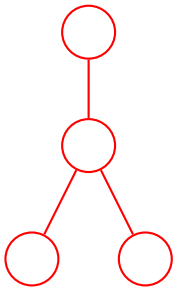
The first possibility for D is:



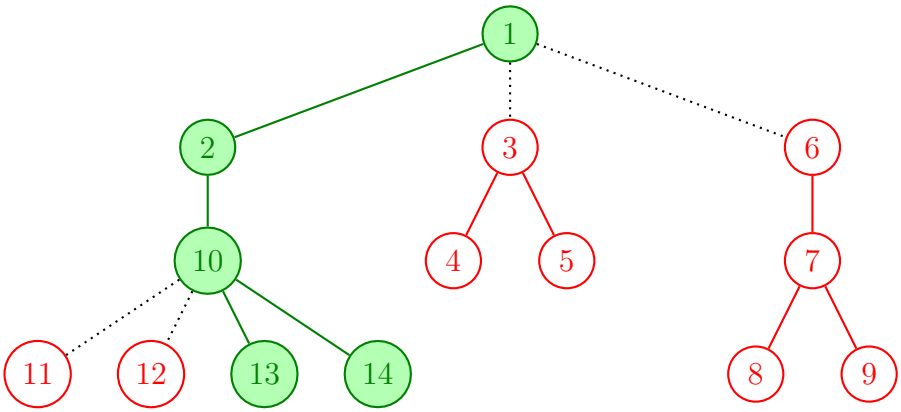
Using this, we can get T the following way:



The second possibility for D is:



Using this, we can get T the following way:



Balanced Integer

Time Limit: 30 seconds

Problem Description
Since the CCO often uses integers, Alice needs to learn about the integers! A positive integer n can be written in base b as the sequence $d_{m-1}d_{m-2}\dots d_1d_0$ if the following hold:

- Each digit d_i is between 0 and $b-1$, inclusive.
- $d_{m-1} > 0$.
- $n = d_{m-1} \times b^{m-1} + d_{m-2} \times b^{m-2} + \cdots + d_1 \times b^1 + d_0 \times b^0$.

For example, the integer 2025 in base 19 is the sequence $(5, 11, 11)$ because $2025 = 5 \times 19^2 + 11 \times 19^1 + 11 \times 19^0$.

An integer n is b-balanced if, when n is written in base b, the average of the digits is $\dfrac{b-1}{2}$. For example, 2025 is 19-balanced because $\dfrac{5+11+11}{3} = 9 = \dfrac{19-1}{2}$.

Alice can easily find integers that are 19-balanced. However, she has trouble finding integers that are balanced in multiple ways. Given B and N, please help Alice find the minimum integer x such that:

- x is b-balanced, for all $2 \le b \le B$.
- $x \ge N$.

Input Specification
The first line of input contains two space-separated integers B and N ($N \ge 1$).

It is guaranteed that the answer does not exceed 10^{18}.

The following table shows how the available 25 marks are distributed:

Marks Awarded	**Bounds on B**	**Bounds on N**
2 marks	$2 \le B \le 7$	$1 \le N \le 10^4$
6 marks	$2 \le B \le 6$	$N = 10^{10}$
2 marks	$2 \le B \le 7$	None
9 marks	$8 \le B \le 11$	$N = 1$
4 marks	$B = 8$	None
2 marks	$9 \le B \le 11$	None

Output Specification
Output the minimum integer x from the problem statement.

Sample Input 1
```
4 100
```

Output for Sample Input 1
```
141
```

Explanation of Output for Sample Input 1
141 in base 2 is 10001101. The average digit is $\frac{1+0+0+0+1+1+0+1}{8} = 0.5 = \frac{2-1}{2}$. Therefore, 141 is 2-balanced.

141 in base 3 is 12020. The average digit is $\frac{1+2+0+2+0}{5} = 1 = \frac{3-1}{2}$. Therefore, 141 is 3-balanced.

141 in base 4 is 2031. The average digit is $\frac{2+0+3+1}{4} = 1.5 = \frac{4-1}{2}$. Therefore, 141 is 4-balanced.

Lastly, $141 \geq 100$.

Sample Input 2
```
7 10000000000
```

Output for Sample Input 2
```
16926961207710
```

Hint

Feel free to use these code snippets as part of your solution.

```
// Important: If x is 0, the result is undefined.
int base_2_length(unsigned long long x) {
  return 64-__builtin_clzll(x);
}

int base_2_sum(unsigned long long x) {
  return __builtin_popcountll(x);
}
```

Restaurant Recommendation Rescue

Time Limit: 2 seconds

Problem Description
A certain aspiring musician K loves going for shabu-shabu! Recently, she's been to N shabu-shabu restaurants, numbered $1, 2, \ldots, N$, following the following algorithm:

1. K keeps an ordered list of recommendations, starting with restaurant 1.
2. On the i-th day, she visits the next recommended restaurant on her list, which recommends her restaurants $R_i = \{r_{i,1}, \ldots, r_{i,\ell_i}\}$.
3. K appends R_i to her list of restaurants to visit.
4. K repeats steps 2-4 until she runs out of recommended restaurants.
5. K writes down the array $A_0, \ldots, A_{N-1}$, where A_i equals the number of restaurants she was recommended on the $(i+1)$-th day. That is, $A_i = |R_{i+1}|$.

It is guaranteed that $\bigcup_{i=1}^{N} R_i = \{2, \ldots, N\}$ and $R_i \cap R_j = \varnothing$ for $i \neq j$, that is, every restaurant, other than the first, will be recommended by exactly one other restaurant.

Once K finishes her list, K's delinquent friend H decides to play a prank on her! She replaces the array $A_0, \ldots, A_{N-1}$ with another array $B_0, \ldots, B_{N-1}$! K thinks that this new array B_i might just be a cyclic shift of her array, so she asks you to determine all possible $0 \leq k < N$ such that $A_i = B_{(i+k) \bmod N}$, for all $0 \leq i < N$ and **any** valid output of her algorithm $A_0, \ldots, A_{N-1}$.

Furthermore, K will then perform Q operations, where for the i-th operation, she swaps B_{x_i}, B_{y_i} and asks you to do the same on the new array. Can you help K see through her friend's prank?

Input Specification
The first line of input will contain two integers, N ($1 \leq N \leq 500\,000$) and Q ($0 \leq Q \leq 300\,000$).

The next line of input will contain N space-separated non-negative integers, $B_0, B_1, \ldots, B_{N-1}$ ($0 \leq B_i < N$), the initial sequence.

The i-th of the next Q lines of input will contain two integers each, x_i and y_i ($0 \leq x_i, y_i < N$ and $x_i \neq y_i$), indicating you are to swap B_{x_i} with B_{y_i}.

The following table shows how the available 25 marks are distributed:

Marks Awarded	**Bounds on N**	**Bounds on Q**
3 marks	$1 \leq N \leq 8$	$Q = 0$
7 marks	$1 \leq N \leq 5\,000$	$Q = 0$
10 marks	$1 \leq N \leq 500\,000$	$Q = 0$
5 marks	$1 \leq N \leq 500\,000$	$0 \leq Q \leq 300\,000$

Output Specification

For each of the $Q + 1$ arrays (including the initial array $B_0, \ldots, B_{N-1}$), let $S = \{k_1, \ldots, k_m\}$ denote the set of integers $0 \leq k_j < N$ such that there exists a valid output $A_0, \ldots, A_{N-1}$ of K's algorithm such that $A_i = B_{(i+k_j) \bmod N}$ for all $0 \leq i < N$. Output, on a single line, the integers m and $\sum_{i=1}^{m} k_i \pmod{998\,244\,353}$, separated by a space.

In particular, if $S = \varnothing$, your output should be `0 0`.

Sample Input

```
5 3
1 2 0 0 1
0 2
1 3
3 2
```

Output for Sample Input

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

Explanation of Output for Sample Input

The array A is $[1, 1, 2, 0, 0]$; it can be shown this is the only valid output of K's algorithm that corresponds to the array $B = [1, 2, 0, 0, 1]$. One input for K's algorithm that yields this array A is:

$$R_1 = \{2\}$$
$$R_2 = \{3\}$$
$$R_3 = \{4, 5\}$$
$$R_4 = \varnothing$$
$$R_5 = \varnothing.$$

After swapping B_0 and B_2, we get the array

$$B = [0, 2, 1, 0, 1].$$

It can be shown the only valid output of K's algorithm that corresponds to this is

$$A = [2, 1, 0, 1, 0].$$

One possible input to K's algorithm that yields this array A is

$$\begin{aligned} R_1 &= \{2, 3\} \\ R_2 &= \{4\} \\ R_3 &= \varnothing \\ R_4 &= \{5\} \\ R_5 &= \varnothing. \end{aligned}$$

Tips for Python (CIW Only) You are recommended to use fast input (for example, `sys.stdin.read()` and `sys.stdout.write()`) if you are attempting the final subtask.

Patrol Robot

Time Limit: 3 seconds

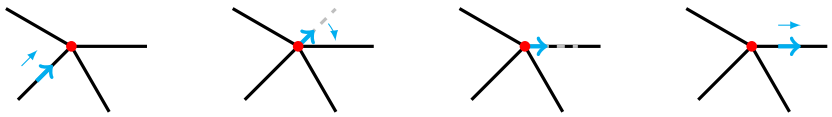
Problem Description

The Coordinate Control Organization has developed an autonomous robot to patrol N distinct important locations on a two-dimensional plane. The i-th location has coordinates (x_i, y_i), and it is guaranteed that no three locations lie on a common line.

To help guide the robot, you may paint some line segments on the ground. Each segment must directly connect two important locations, and no two segments may intersect, except possibly at their endpoints.

The robot will begin its patrol at the midpoint of an arbitrary segment, facing towards one of its endpoints. It will move indefinitely according to the following procedure:

- As long as the robot is in the interior of a segment, it will move forward, towards a segment endpoint.
- When the robot reaches an important location, it will initially be facing directly away from the segment it just traversed. The robot will turn right/clockwise until its line of vision is aligned with a segment that leads away from the current location. The robot will then begin moving along this new segment.



Your task is to paint the segments in such a way that, no matter where the robot starts, it is guaranteed to visit every important location infinitely often. It can be proven that this is always possible.

Input Specification

The first line of input contains a single integer N ($2 \leq N \leq 2\,000$), the number of important locations.

The next N lines of input each contain two space-separated integers, x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$), the coordinates of the i-th important location.

It is guaranteed that all N important locations are distinct and no three lie on a common line.

The following table shows how the available 25 marks are distributed:

Marks Awarded	Additional Constraints
2 marks	$2 \leq N \leq 4$
4 marks	$2 \leq N \leq 8$
3 marks	$3 \leq N$ and the N points are the vertices of a convex polygon in some order.
7 marks	The N points form a convex polygon with $N-1$ vertices plus an additional point inside the polygon.
9 marks	None

Output Specification
On the first line, output a positive integer M, the number of line segments you paint on the ground.

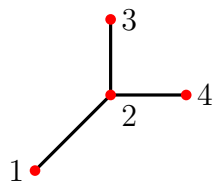
The next M lines of output should each contain two space-separated integers, u_i and v_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i$), denoting that you paint a line segment between the u_i-th and v_i-th important locations.

If there are multiple acceptable answers, output any of them.

Sample Input 1
```
4
0 0
1 1
1 2
2 1
```

Output for Sample Input 1
```
3
1 2
2 3
2 4
```

Explanation of Output for Sample Input 1
The important locations and painted segments are shown in the following figure:



No matter where the robot starts, it will visit every important location infinitely many times. For example, if the robot starts in the middle of the segment between locations 1 and 2, facing towards location 2, the locations the robot will visit in order are:

$$\underline{2, 4, 2, 3, 2, 1}, 2, 4, \ldots,$$

where the underlined portion is repeated infinitely.

Sample Input 2

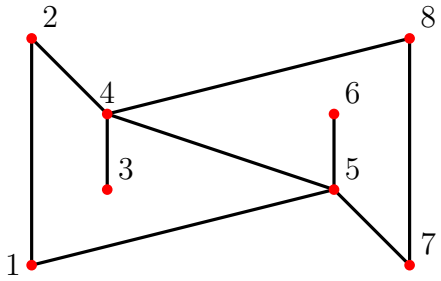
```
8
0 0
0 3
1 1
1 2
4 1
4 2
5 0
5 3
```

Output for Sample Input 2

```
9
1 2
2 4
4 8
8 7
7 5
5 1
3 4
4 5
5 6
```

Explanation of Output for Sample Input 2

The important locations and painted segments are shown in the following figure:



No matter where the robot starts, it will visit every important location infinitely many times.

For example, if the robot starts in the middle of the segment between locations 4 and 5, facing towards location 5, the locations the robot will visit in order are:

$$5, \underline{7, 5, 6, 5, 1, 2, 4, 3, 4, 8}, 7, 5, \ldots,$$

where the underlined portion is repeated infinitely. Note that it is not necessary for every segment to be traversed infinitely many times; the solution is valid as long as every location is visited infinitely many times.

Shopping Deals

Time Limit: 5 seconds

Problem Description

You are shopping from a store that sells a total of M items. The store layout can be modelled as a two-dimensional plane, where the i-th item is located at the point (x_i, y_i) and has a price of p_i.

The store offers N shopping deals. The i-th shopping deal is specified by a point (a_i, b_i), and for a cost of c_i, you can obtain one of every item within exactly one of the following four regions of your choice:

- The region of points (x, y) such that $x \leq a_i$ and $y \leq b_i$.
- The region of points (x, y) such that $x \leq a_i$ and $y \geq b_i$.
- The region of points (x, y) such that $x \geq a_i$ and $y \leq b_i$.
- The region of points (x, y) such that $x \geq a_i$ and $y \geq b_i$.

Each shopping deal can only be used at most once. Items can also be purchased individually by paying their respective price p_i.

You want to obtain at least one of each item in the store. Find the minimum total cost you must pay to do so.

Input Specification

The first line of input contains two space-separated integers N and M.

The next N lines of input each contain three space-separated integers, a_i, b_i, and c_i ($-10^9 \leq a_i, b_i \leq 10^9, 1 \leq c_i \leq 10^9$).

The next M lines of input each contain three space-separated integers, x_i, y_i, and p_i ($-10^9 \leq x_i, y_i \leq 10^9, 1 \leq p_i \leq 10^9$).

The following table shows how the available 25 marks are distributed:

Marks Awarded	Bounds on N	Bounds on M	Additional Constraints
1 mark	$1 \leq N \leq 8$	$1 \leq M \leq 20$	None
3 marks	$1 \leq N \leq 70$	$1 \leq M \leq 20$	None
3 marks	$1 \leq N \leq 70$	$1 \leq M \leq 70$	None
4 marks	$1 \leq N \leq 100$	$1 \leq M \leq 100\,000$	No two points (a_i, b_i) or (x_j, y_j) have the same x or y-coordinate.
2 marks	$1 \leq N \leq 100$	$1 \leq M \leq 100\,000$	None
8 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 100\,000$	No two points (a_i, b_i) or (x_j, y_j) have the same x or y-coordinate.
4 marks	$1 \leq N \leq 1\,000$	$1 \leq M \leq 100\,000$	None

Output Specification
On a single line, output the minimum total cost that you must pay to obtain at least one of each item.

Sample Input
```
2 4
1 1 3
3 3 13
0 0 2
0 2 5
2 0 4
2 2 3
```

Output for Sample Input
```
12
```

Explanation of Output for Sample Input
Use the first shopping deal on the region $\{(x, y) \mid x \leq 1, y \geq 1\}$ to obtain the second item. Then, purchase items 1, 3, and 4 individually. The total cost is $3 + (2 + 4 + 3) = 12$.