**JOI 2016-2017 Qualification contest**

Authors: JCIOI (Japanese Committee for the IOI)

Translators: Jasmine Chua, France-ioi

License: Creative Commons Attributions-ShareAlike 4.0 International License (CC BY-SA 4.0)

**Microwave**

**QUESTION**

To prepare for a meal, JOI heats up a piece of meat at A°C in the microwave until it reaches B°C. Meat freezes at temperatures <0°C. When the temperature is >0°C, the meat is not frozen. When the temperature is at 0°C, the meat can be frozen or not frozen.

JOI has the following hypothesis about the time it takes for meat to heat up. He used it to estimate how long it will take to heat up the meat.

* When the meat is frozen, and its temperature is <0°C: The meat heats up by 1°C in C seconds.
* When the meat is frozen, and its temperature is at 0°C: The meat defrosts in D seconds and becomes unfrozen.
* When the meat is not frozen: The meat heats up by 1°C in E seconds.

Following these estimates, how many seconds does it take to heat the meat up to B°C?

**INPUT**

The input contains 5 rows, with 1 integer written in each row.

In the 1st row, the meat's original temperature, A, is written.

In the 2nd row, the target temperature, B, is written.

In the 3rd row, the time it takes frozen meat to heat up by 1°C, C, is written.

In the 4th row, the time it takes frozen meat to defrost, D, is written.

In the 5th row, the time it takes not frozen meat to heat up by 1°C, E, is written.

The original temperature A is -100 < A < 100, the target temperature B is 1 < B < 100, while A ≠ 0 and A < B.

All times taken to heat up the meat, C, D, and E, are 1 < C/D/E < 100.

**OUTPUT**

Output one line, containing the number of seconds it takes to heat the meat up to B°C.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

-10

20

5

10

3

**OUTPUT 1**

120

**INPUT 2**

35

92

31

50

11

**OUTPUT 2**

627

In INPUT EXAMPLE 1, the meat is originally frozen at -10°C. The time taken is as follows:

* To heat up from -10°C to 0°C: 5 × 10 = 50 seconds.
* To defrost at 0°C: 10 seconds.
* To heat up from 0°C to 20°C: 3 × 20 = 60 seconds.

Therefore, the total time taken is 120 seconds.

In INPUT EXAMPLE 2, the meat is not frozen. Therefore, it takes 627 seconds to heat the meat up from 35°C to 92°C.

※Save each example's input/output data as a file by right-clicking the links.

**Point Card**

**QUESTION**

JOI Shopping Mall has a point card system. Each point card has 2N squares. When a purchase is made, the customer can draw a lottery, and the result of 'Win' or 'Miss' is stamped onto a square. Each square can only receive 1 stamp. Point cards with ≥N squares stamped with 'Win' can be exchanged for a prize. In addition, stamps on the point card can be overwritten at 1 yen each.

JOI has M point cards, all of which have all 2N squares stamped. Point card i, where 1 ≦ i ≦ M, has Ai 'Win' stamps and Bi 'Miss' stamps on it. JOI wants to get ≥ (M - 1) prizes.

What is the minimum amount of money JOI must spend to get ≥ (M - 1) prizes?

**INPUT**

The input has (M + 1) rows.

In the 1st row, 2 integers N and M, where 1 ≦ N ≦ 1000 and 1 ≦ M ≦ 1000, are written with a space between them. This represents each point card having 2N squares, and JOI having M point cards.

For the next M rows, in the ith row, where 1 ≦ i ≦ M, 2 integers Ai and Bi, where 0 ≦ Ai ≦ 2N, 0 ≦ Bi ≦ 2N, and Ai + Bi = 2N, are written. This represents point card i having Ai 'Win' stamps and Bi 'Miss' stamps on it.

**OUTPUT**

Output the minimum amount of money JOI must spend to get ≥(M - 1) prizes in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

4 5

1 7

6 2

3 5

4 4

0 8

**OUTPUT 1**

4

**INPUT 2**

5 4

5 5

8 2

3 7

8 2

**OUTPUT 2**

0

In INPUT EXAMPLE 1, JOI needs to overwrite 3 'Miss' stamps on point card 1 to 'Win' stamps, then overwrite 1 'Miss' stamp on point card 3 to a 'Win' stamp. After spending 4 yen, he has 4 (= 5 - 1) point cards to exchange for a prize, so that is the minimum amount of money needed.

In INPUT EXAMPLE 2, he already has 3 (= 4 - 1) point cards to exchange for a prize, so there is no need to overwrite any cards.

※Save each example's input/output data as a file by right-clicking the links.

**Refreshment Area**

**QUESTION**

An international programming contest is being held in Japan, and the venue is being constructed now. The venue is represented as grid, with N squares in the North-South direction, and M squares in the East-West direction. Several squares have equipment to be used in the contest installed on them.

In order to let the contestants take a break during competition, one refreshment area where they can have a snack and drink will be set up in the venue. The area must take up D consecutive squares in either the North-South and East-West direction. However, the refreshment area cannot be located on squares where equipment has been installed.

Create a program to find out how many possible ways there are to set up a refreshment area in the venue.

**INPUT**

The input has (1 + N) rows.

In the 1st row, 3 integers N, M, and D, where 1 ≦ N ≦ 100, 1 ≦ M ≦ 100, and 2 ≦ D ≦ 100, are written with a space between them. This represents the venue having N squares in the North-South direction, M squares in the East-West direction, and the refreshment area taking up D consecutive squares in either the North-South and East-West direction.

For the next N rows, strings composed of M characters are written to show the venue's information. The jth character in the ith row, where 1 ≦ i ≦ N and 1 ≦ j ≦ M, represents the situation of the square that is ith from the North and jth from the West. Each character can be a '#' or '.'. '#' represents a square with equipment installed, while '.' represents a square with no equipment installed.

**OUTPUT**

Output the number of possible ways to set up a refreshment area in the venue.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

3 5 2

...#.

#...#

....#

**OUTPUT 1**

12

**INPUT 2**

4 7 5

.#.....

.....##

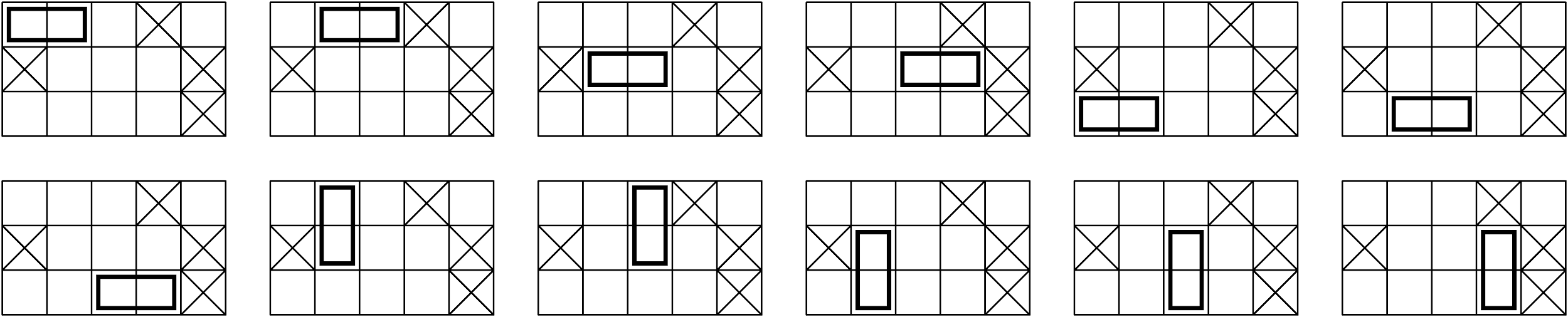
.......

#......

**OUTPUT 2**

7

In INPUT EXAMPLE 1, there are a total of 12 possible ways to set up a refreshment area, as illustrated in the image below.



※Save each example's input/output data as a file by right-clicking the links.

**Plush Toys**

**QUESTION**

JOI works in a toy store. Today, he is in-charge of tidying up the plush toy section.

In the plush toy section, N plush toys are placed in a row from left to right. The shelf is divided into N sections by partitions, and each section can hold 1 plush toy. The toy store sells a total of M types of plush toys, and each has a serial number from 1 to M. All the N plush toys on the shelf belong to one of these M types. There is at least 1 of each type.

To make the shelf look good, JOI wants to place plush toys of the same type next to each other. He rearranged the plush toys using the following method:

* Several plush toys will be chosen and removed from the shelf. Those that are not removed will remain at their original positions.
* Plush toys that are removed will be placed back into the empty sections in any order.

After rearranging, plush toys of the same type must be placed next to each other. Create a program to find out the minimum number of plush toys that needs to be removed from the shelf.

**INPUT**

The input has (1 + N) rows.

In the 1st row, 2 integers N and M, where 1 ≦ N ≦ 100000 and 1 ≦ M ≦ 20, are written with a space between them. This represents that there are N plush toys of M types.

For the next N rows, integers ranging from 1 to M are written. The ith row, where 1 ≦ i ≦ N, represents the type of the plush toy in the ith section from the left. Ensure that each type of plush toy appears at least once.

**OUTPUT**

Output the minimum number of plush toys that needs to be removed from the shelf in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

7 2

1

2

2

2

1

2

1

**OUTPUT 1**

2

**INPUT 2**

12 4

1

3

2

4

2

1

2

3

1

1

3

4

**OUTPUT 2**

7

In INPUT EXAMPLE 1, the type of plush toy on the shelf, starting from the left, is 1, 2, 2, 2, 1, 2, 1. To get the minimum number of plush toys that needs to be removed, the 1st and 6th plush toys from the left should be removed. The type 2 plush toy should be replaced in the 1st section, and the type 1 plush toy should be replaced in the 6th section. In this situation, the number of plush toys that needs to be removed is 2.

※Save each example's input/output data as a file by right-clicking the links.

**Ridge**

**QUESTION**

JOI Caldera is a beautiful formation with a great view that attracts many hikers. The view from places known as ridges is especially breathtaking.

JOI Caldera takes up a rectangular space, measuring H kilometers from North to South, and W kilometers from East to West. The space is segregated into a grid with each square measuring 1 x 1 kilometer, making up a total area with H×W squares, called 'regions'. The altitude throughout each region is uniform. In addition, each region must have a different altitude.

If it rains on a certain region, the rain will flow into the 4 regions adjacent to it, to the North, South, East, and West, if those regions have an altitude lower than it. If none of the adjacent regions are at a lower altitude, the rainwater will stay in the original region. Likewise, rainwater that flows into a region can flow out to any adjacent regions that are lower in altitude. JOI Caldera is surrounded by an outer ring of steep cliffs, so rainwater cannot flow out of JOI Caldera.

A region is known as a 'ridge' if any rain that falls on it eventually flows and ends up in multiple other regions. Create a program to find out how many ridges there are.

**INPUT**

The input has (1 + H) rows.

In the 1st row, 2 integers H and W, where 1 ≦ H ≦ 1000 and 1 ≦ W ≦ 1000, are written with a space between them. This represents that JOI Caldera measures H kilometers from North to South, and W kilometers from East to West.

For the next H rows, W integers are written with a space between them, to represent altitude.

The jth integer in the ith row, where 1 ≦ i ≦ H and 1 ≦ j ≦ W, is (Mi,j), where 1 ≦ Mi,j ≦ H×W, and represents the altitude of the region in the ith row, jth column. If (i,j) ≠ (k,l), then Mi,j ≠ Mk,l.

**OUTPUT**

Output the number of ridges in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

3 3

2 9 4

7 5 3

6 1 8

**OUTPUT 1**

4

**INPUT 2**

3 5

5 3 8 2 14

9 10 4 1 13

12 7 11 6 15

**OUTPUT 2**

4

In INPUT EXAMPLE 1, there are four ridges, at altitudes 5, 7, 8, and 9. For example, when rain falls on the region with altitude 9, the rainwater will eventually flow into three regions, with altitude 1, 2, and 3. Therefore, the region with altitude 9 is a ridge. In contrast, when rain falls on the region with altitude 6, the rainwater will eventually flow only to the region with altitude 1. Therefore, the region with altitude 6 is not a ridge.

In INPUT EXAMPLE 2, there are four ridges, at altitudes 8, 10, 11, and 12.

※Save each example's input/output data as a file by right-clicking the links.

**Snake JOI**

**QUESTION**

Snake JOI is lost in a huge mansion. It needs to escape before it is found by the mansion's owner.

There are N rooms in the mansion, numbered 1, 2, ..., N. There are also M hallways, and the ith hallway, where 1 ≦ i ≦ M, connects rooms Ai and Bi. JOI is free to move in either direction along these corridors and moving down corridor i takes Di minutes. There is no other method to travel between rooms.

Each room in the mansion is set at a certain temperature, which can be too cold, just right, or too hot for JOI. As JOI cannot adjust to rapid temperature changes, it needs to wait at least X minutes after exiting a cold room before it can enter a hot room. Similarly, it needs to wait at least X minutes after exiting a hot room before it can enter a cold room.

When moving, JOI needs to exit a room immediately after entering it. Also, it cannot suddenly make a turn in a corridor and head back, or use more than Di minutes to travel through corridor i. However, it can return to the same room or travel down the same corridor more than once.

Now, JOI is in room 1. This room is too cold for JOI. JOI needs to enter room N where the exit is, to escape the mansion.

What is the minimum amount of time JOI needs to escape the mansion?

**INPUT**

The input has (1 + N + M) rows.

In the 1st row, 3 integers N, M, and X, where 2 ≦ N ≦ 10000, 1 ≦ M ≦ 20000, and 1 ≦ X ≦ 200, are written with a space between them. This represents the mansion having N rooms with M corridors, and the X minutes it takes for JOI to adjust to temperature changes.

For the next N rows, in the ith row, where 1 ≦ i ≦ N, integer Ti, where 0 ≦ Ti ≦ 2, is written to represent temperature. In room i, Ti = 0 represents a room that is too cold, Ti = 1 represents a room that is just right, and Ti = 2 represents a room that is too hot. Ensure that T1 = 0.

For the following M rows, in the jth row, 3 integers Aj, Bj, and Dj, where 1 ≦ Aj < Bj ≦ N and 1 ≦ Dj ≦ 200, are written with a space between them. This represents corridor j connecting rooms Aj and Bj, and the Dj minutes it takes to travel between them. Note that there may be more than one corridor connecting the same two rooms.

Ensure that the input data allows JOI to escape the mansion.

**OUTPUT**

Output the minimum amount of time JOI needs to escape the mansion in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

8 10 4

0

1

1

2

1

1

2

0

1 2 1

1 3 1

2 3 3

2 4 5

3 4 1

4 5 1

5 6 1

5 8 1

1 7 2

7 8 2

**OUTPUT 1**

9

**INPUT 2**

15 25 4

0

1

1

0

2

1

0

1

1

2

0

0

1

0

1

8 11 1

7 10 1

12 14 1

3 8 1

1 5 1

3 9 1

3 8 1

1 5 1

6 15 1

11 12 1

2 14 1

7 10 1

11 12 1

5 13 1

2 8 1

1 4 1

2 11 1

5 6 1

1 13 1

6 12 1

5 10 1

9 13 1

4 10 1

3 12 1

7 13 1

**OUTPUT 2**

6

In INPUT EXAMPLE 1, the minimum amount of time is taken when traveling in this sequence: 1 → 2 → 3 → 4 → 5 → 6 → 5 → 8.

In INPUT EXAMPLE 2, there are multiple corridors connecting two rooms (e.g. room 1 and room 5).

※Save each example's input/output data as a file by right-clicking the links.