

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem A. Island Network

An archipelago consists of several groups of islands and the main island from each group has access to Internet. Now it is time to provide Internet to the smaller islands of each group by connecting them with cables to the main island. The company which lays down the cables would like to minimize the expenses for building the network, given that the price of connecting one island from a group to another is equal to the distance in kilometers between the two islands. The installation of the cables begins **at the same time** and has the same speed of one kilometer per month. Given that, each island has a time t_i required to get Internet access.

Each island has m_i inhabitants, which will get access to the Internet. The provider company wants to calculate the average time for implementing the project by using the formula:

$$\frac{\sum m_i t_i}{\sum m_i}$$

Input.

The standard input will contain the data for all groups of islands in the archipelago. The first line of data for each group will contain the number of islands n , followed by n lines of data for each island – the coordinates x_i and y_i of the island in kilometers, and m_i – the number of its inhabitants. The main island's data are on the first line. After the data for the last group, there will be a line containing a zero.

Limitations.

$3 \leq n \leq 100$, $100 \leq m_i \leq 5000$, $0 \leq x_i, y_i \leq 1000$

Output.

For each group of islands the program should output a line, starting with the sequence number of the group (starting from 1) and the average time with two decimal points.

Example:

Input.

```
5
2 7 1000
2 10 200
2 15 200
7 2 500
2 10 100
0
```

Output.

```
1 2.72
```

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem B. Subsequences

You have a sequence of n whole numbers. Check whether a certain number can be obtained by summing the values of a subsequence of the given sequence.

Input.

For each test, the standard input contains: the length of the sequence n , the values of the members in that sequence, the number of checks m and the values to be checked.

Limitations.

$2 \leq n, m \leq 100$

The values of the members and the values to check are in the range $[0, 100000]$.

Output.

For each test you should output a sequence of m strings "yes" and "no", based on whether the respective value to check can be obtained by summing the members of a subsequence of the given sequence.

Example:

Input.

```
4
2 4 5 9
3
15 10 5
```

Output.

```
yes no yes
```

Explanation:

For subsequence 2, 4, 9 we get $2 + 4 + 9 = 15$. No subsequence has members whose sum is 10. The sum 5 is obtained by the subsequence with only member 5.

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem C. Che)(mate

Harry Caspary decided to play some chess, but when he took the chess board, he was amused to see that his son has written a number in each cell. Harry remained calm and instead of getting angry, came up with a new game. He decided to switch each prime number of the board with an opening parentheses – (, and the rest with a closing one –). However, to introduce some rules to the game, he decided to adhere to the following limitations:

1. Only one chess piece can participate in the game – a knight.
2. There is a starting position of the knight on the board – number of row and number of column, where it initially stands.
3. There is a final position of the knight on the board – number of row and number of column, which should be reached.
4. The knight keeps moving as in chess – in the shape of the letter Γ , until it reaches the final position, without stepping on the same square twice.
5. To play the game, one should consider only the square on which the knight has been placed (stepped).

The goal of the game was to determine whether by moving on the board, the knight can “gather” a sequence of parentheses, forming a correct expression of opening and closing parentheses. A correct expression is one of the form $()$, $((()()))$, and incorrect $-)()$, $(()$, $((()$, i.e. for each of opening parentheses there should be a corresponding closing one. The expression stops when the knight reaches the designated final position on the board.

Write a program, which receives as input a matrix of whole numbers a_{ij} ($N \times N$), builds the board of Harry Caspary by changing the prime numbers with opening parentheses ' (' and the rest with closing ')' and determines the length of the longest correct expression of parentheses, gathered by all possible moves of the knight in the shortest path from a given position (M_1, N_1) to another position (M_2, N_2) , where M_1 is the row number and N_1 is the column number, as for the final position M_2 and N_2 .

Input.

For each test the first line of the stdin will contain a number N – the size of the matrix, and the next line will contain the numbers M_1 and N_1 – number of row and number of column of the starting position of the knight. The third line will contain M_2 and N_2 – number of row and number of column of the final position of the knight. The next N lines contain the rows of the matrix – whole numbers, separated by space. The input will contain multiple tests.

Limitations.

$$3 \leq N \leq 13;$$

$$1 \leq N_i, M_i \leq N, i = 1, 2;$$

$$0 \leq a_{ij} \leq 10^5, i, j = 1, 2, \dots, N.$$

NEW BULGARIAN UNIVERSITY
 Department of Informatics
 XXXIII National Student Programming Contest
 May 8, 2021

Output.

For each test, on a separate line output the length of the longest correct expression of parentheses, gathered by all possible moves on the knight's path. If there is no solution, output 0.

Example:

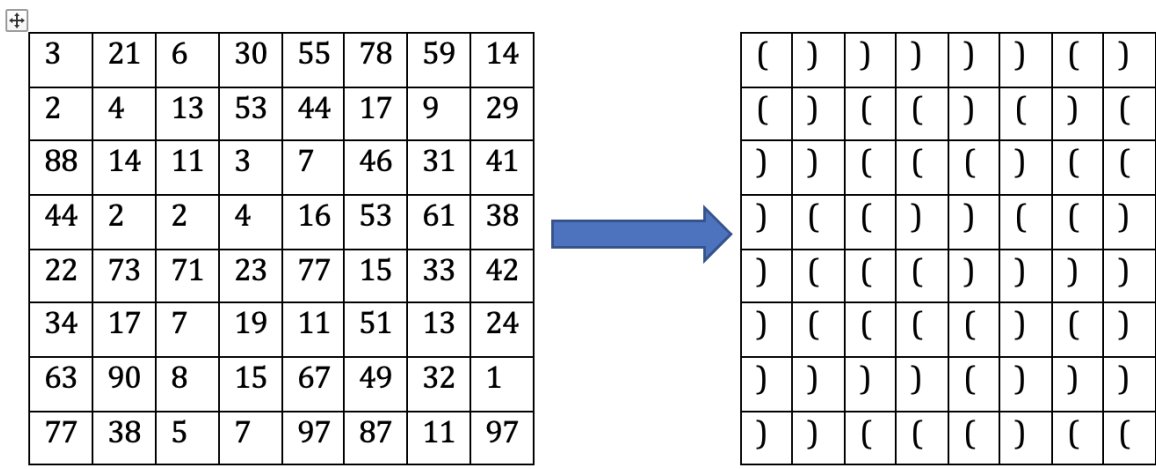
Input.

```
8
7 1
8 3
3 21 6 30 55 78 59 14
2 4 13 53 44 17 9 29
88 14 11 3 7 46 31 41
44 2 2 4 16 53 61 38
22 73 71 23 77 15 33 42
34 17 7 19 11 51 13 24
63 90 8 15 67 49 32 1
77 38 5 7 97 87 11 97
```

Output.

12

Explanation of the example.



The gathered expression is `) ((() () ())) ((` and the longest correct expression inside it is `((() () ()))`, having a length 12. The knight has passed through the following squares: (7, 1), (5, 2), (6, 3), (8, 3), (3, 1), (3, 3), (4, 4), (6, 4), (7, 3), (4, 2), (5, 1), (8, 2), (5, 5), (7, 5), (8, 4).

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem D. Crossings

You are given horizontal and vertical lines in the plane, which have whole number end points. Find the total number of cross points of a horizontal and a vertical line.

Input.

For each test, the standard input contains: the number of lines and coordinates of their ends. The input contains multiple tests.

Limitations.

The number of lines is below 2000.

The line coordinates are whole numbers in the interval $[-10^9, 10^9]$.

The lengths of the lines are greater than 0.

Output.

For each test, output on a separate line the required number.

Example:

<i>Input.</i>	<i>Output.</i>
4	1
0 0 5 0	1
1 1 1 -1	
0 2 5 2	
-1 0 3 0	
2	
0 0 0 1	
0 0 1 0	

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem E. Windows

You are given a square screen of pixels with $M + 1$ rows and $N + 1$ columns, numbered respectively from 0 to M and 0 to N . A window is a square of pixels with an upper left corner at coordinates (x, y) , having r rows and c columns, positioned entirely on the screen. The windows placed on the screen cannot have common pixels. Write a program that controls the placement of windows on the screen. In the beginning there are no windows on the screen. The program should process the following commands:

- $O\ x\ y\ r\ c$ – opens a window with on the screen with parameters $x\ y\ r$ and c .
- $C\ x\ y$ – closes the window that contains the pixel with coordinates (x, y) .
- $R\ x\ y\ r\ c$ – changes the size of the window that contains the pixel with coordinates (x, y) to r rows and c columns.
- $M\ x\ y\ r\ c$ – moves the window that contains the pixel with coordinates (x, y) vertically by r pixels or horizontally by c pixels, where exactly one of r and c will be different than 0. A positive c means movement down, and a negative one – up. A positive r means movement to the right, and negative – to the left.

During the execution of the commands, there might be exceptions, which should be reported by the program in the following format:

`<number of command> <type of command> <type of exception>`

as the numbering of commands for each test starts from 1, the type is either O, C, R or M, and the possible exceptions are:

- `Doesn't exist` – when for a command C, R or M the given window is not on the screen.
- `Can't fit` – when for a command O, R or M the resulting window cannot fit on the screen – part of it is outside of the screen or overlaps with another window.

Input.

The program should solve several test cases in one run. For each test, the standard input will contain the parameters M and N of the screen and the number of commands Q . Each of the next Q lines will contain one command. After the last test there will be three zeroes.

Limitations.

$100 \leq M \leq 1000$, $100 \leq N \leq 1000$, $5 \leq Q \leq 100$

Output.

For each command that cannot be executed, the program should output the corresponding exception message, and after executing all commands for a given test – the number of open windows remaining on the screen.

NEW BULGARIAN UNIVERSITY
Department of *Informatics*
XXXIII National Student Programming Contest
May 8, 2021

Example:

Input.

```
320 200 12
O -5 -5 50 50
O 50 50 10 10
R 60 60 300 300
O 70 55 10 10
O 90 50 10 10
R 70 60 40 40
R 55 55 40 40
R 55 55 15 15
M 0 0 40 0
C 55 55
C 300 200
M 95 55 0 -100
0 0 0
```

Output.

```
1 O Can't fit
3 R Can't fit
9 M Doesn't exist
11 C Doesn't exist
12 M Can't fit
2
```

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem F. Grasshopper

Given are 4 whole numbers: k , d_1 , d_2 , n and a sequence of numbers with length n . A grasshopper stands on the natural number line at the point of number k . From that position it starts to make jumps along the number line, with length d_1 or d_2 in both distances, i.e. it can go to the points $k + d_1$, $k + d_2$, $k - d_1$ and $k - d_2$. Write a program that finds out which points from the given sequence are reachable by the grasshopper in an arbitrary number of jumps.

Input.

For each test case the standard input contains two lines. The first line contains the numbers k , d_1 , d_2 and n , separated by space, and the second – the sequence of numbers, again separated by space. The input contains multiple tests.

Limitations.

$2 \leq n \leq 700$

$1 \leq k, d_1, d_2 \leq 1000$

The elements of the sequence are in the interval $[-1000, 1000]$.

Output.

For each test in the standard input, output a sequence of zeroes and ones (separated by space), which shows which elements of the input sequence are reachable by the grasshopper.

Example:

Input.

```
10 4 6 3
10 15 20
8 3 2 2
9 4
10 4 6 10
12 33 40 100 -200 201 1000 2 4 6
```

Output.

```
1 0 1
1 1
1 0 1 1 1 0 1 1 1 1
```


NEW BULGARIAN UNIVERSITY
Department of *Informatics*
XXXIII National Student Programming Contest
May 8, 2021

Problem G. Signals

The transmission of binary strings over a data transmission channel involves alternating ones and zeroes. If a string of ones and zeroes contains more ones, which are coded with higher voltage than zeroes on the channel, we call this string heavy, as it can overload the equipment. Your task is, for a given binary string to find the number of heavy substrings.

Input.

A line of the standard input will contain the number of tests T . Each test case will be comprised of a line, containing a binary string with length N .

Limitations.

$4 \leq N \leq 1000000$

Output.

For each test, the program should output on a separate line the number of heavy substrings for the given string.

Example:

<i>Input.</i>	<i>Output.</i>
2	4
0011	5
0110	

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem H. Jeep

Chris's parents bought him his first car: a monster jeep! Chris found out that despite having a car, which can smash all other cars, parking an automobile of the size of 4 normal cars can be very complicated. His friend Bobby works in the company, responsible for city parking lots. He periodically sends Chris a map of the city, with designated parking places, on which the taken spots are marked. The map can be represented as a table with R rows and C columns. Each table cell can contain a building (marked with the symbol "#"), a parked car (marked with "X") or a free parking spot (marked with "."). Chris's monster jeep is huge and requires a square of 2×2 cells to be parked. Help Chris find out the number of possible parking places, grouped by the number of cars, which should be moved in order to park at a given place. We are interested only in the number of cars to be moved away from the parking place, and not in the number of cars which should move along the jeep's path to the parking place. Chris cannot park on a building, and it cannot pass through buildings.

Input.

The first line of the standard input contains a number N – the number of test. The first line of each test contains two whole numbers, R and C , the number of rows and columns of the map. Each from the second to the R -th lines of the test contain C characters, which can be among "#", "X" and "." "X" is the uppercase Latin letter X.

Limitations.

$2 \leq R, C \leq 50$

Output.

For each test, the program should output five lines: the total number of cars which should be moved in order for Chris to park, if 0 cars need to be moved (on the first line), if 1 car (on the second line), if 2 cars (on the third line), if 3 cars (on the fourth line), if 4 cars (on the fifth line).

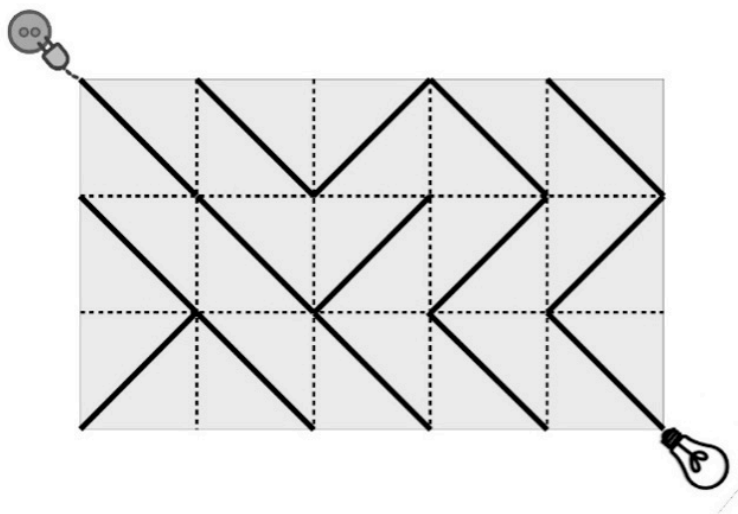
NEW BULGARIAN UNIVERSITY
Department of *Informatics*
XXXIII National Student Programming Contest
May 8, 2021

Example:

<i>Input.</i>	<i>Output.</i>
3	1
4 4	1
#..#	2
..X.	1
..X.	0
#XX#	9
4 4	0
....	0
....	0
....	0
....	2
4 5	1
..XX.	1
..#XX.	0
..#..	1
.....	

Problem I. Lights On

During one of his visits in the National Polytechnic Museum, Peter comes across an interesting exhibit. Inside a rectangular frame, there are ceramic tiles, tightly arranged next to each other. They form a rectangular frame that has N rows and M columns. Two (of the four) opposite angles of each tile are connected to a conductor. There is a power source connected to the upper left corner of the frame. There is a lamp connected to the lower right corner of the frame. The lamp is on only when there is a path of conductors connecting the power source with the lamp.



On the Figure above, the lamp is turned off. To turn it on, some of the tiles need to be rotated to 90 degrees (regardless of the direction), so the conductors form a path from the power source to the lamp. If any of the tiles in the fourth column of the Figure is rotated to 90 degrees, the power source and the lamp will be connected and the lamp will be on. Peter is trying to find the minimal number of rotations of tiles for lighting the lamp. Help him by writing a program "lamp" which uses the configuration of the tiles and determines the minimal number of 90-degree rotations to switch on the lamp.

Input.

The first line of the standard input contains a whole number T , which determines the number of tests. The first line of each test contains two whole numbers – N and M – the number of rows and columns of tiles in the frame. Each of the next N lines contains adjacent symbols ' \backslash ' or ' $/$ ', without spaces between, which show the direction of the conductor that connects the opposite angles of the corresponding tile.

Limitations.

$1 \leq N, M \leq 500$

NEW BULGARIAN UNIVERSITY
Department of *Informatics*
XXXIII National Student Programming Contest
May 8, 2021

Output.

For each test, print on the standard output a whole number: the minimum number of tiles that should be rotated by 90 degrees, to switch the lamp on. If this is not possible, output "Impossible".

Example:

<i>Input.</i>	<i>Output.</i>
2	1
3 5	3
\\/\	
\\//	
/\	
3 3	
///	
//\	
///	

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem J. Zigzag

A finite sequence of whole numbers is called a Zigzag, if every element in the sequence, except the first and the last, is either greater than or less than its two neighbors. Write a program to determine the longest part in a sequence, which is Zigzag.

Input.

The standard input contains sequences of numbers – each on a separate line, with a space between the numbers.

Limitations.

All numbers are in the interval $[-1000, +1000]$.

The length of the sequence is at least 3 and at most 10^6 .

Output.

For each sequence, output on a separate line the length of the longest Zigzag part of that sequence.

Example:

<i>Input.</i>	<i>Output.</i>
2 30 2 3 2 3 2 2	7
1 2 3 4 5 6 7 81	0
1 1 1 2 1 1 3 2 3 3	4
1 2 2 1	0

Example Explanation:

2 30 2 3 2 3 2 is a Zigzag sequence.

No part is a Zigzag sequence.

1 2 1 and 1 3 2 3 are Zigzag sequences.

No part is a Zigzag sequence.

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem K. Input-output

You have a finite sequence of strings, generated by a software system, processed and sent away. The string length determines the cycles for which it is processed, i.e. after how many system cycles it can be sent. At each cycle, a new string enters the system and one already processed string can be sent away. If more than one string is ready to be sent, the shortest one is sent; if there are more than one that are shortest, the lexicographically smallest one is sent.

Input.

The standard input receives a sequence of strings – strings which enter the system at each cycle.

Limitations.

The strings contain lowercase Latin letters and are no longer than 100 characters.

Output.

For each input string, you should output a line containing the input string and the string that is sent on this cycle. If there is no string to send, output `-`. The output ends when there are no strings that enter the system and no strings that need to be sent. The exact format of the output is given in the example below.

Example:

Input.

```
axxxx xb abc da bb c
```

Output.

```
1 in: axxxx out:-
2 in: xb out:-
3 in:abc out:-
4 in:da out:xb
5 in:bb out:-
6 in:c out:da
7 in:- out:c
8 in:- out:bb
9 in:- out:abc
10 in:- out:axxxx
```

Example explanation:

At the first cycle the string `axxxx` enters, which will be processed in 5 cycles, therefore the earliest it can be sent is at cycle 6. No ready strings to send.

At the second cycle the string `xb` enters, which will be processed in 2 cycles, therefore the earliest it can be sent out is at cycle 4. No ready strings to send.

NEW BULGARIAN UNIVERSITY
Department of *Informatics*
XXXIII National Student Programming Contest
May 8, 2021

At the third cycle a string `abc` enters, which will be processed in 3 cycles, therefore the earliest it can be sent out is at cycle 6. No ready strings to send.

At the fourth cycle a string `da` enters, which will be processed in 2 cycles, therefore the earliest it can be sent is at cycle 6. String `xb` is sent out.

At the fifth cycle enters the string `bb`, which will be processed in 2 cycles, therefore the earliest it can be sent is at cycle 7. No ready strings to send.

At the sixth cycle enters the string `c`, which will be processed in 1 cycle, therefore the earliest it can be sent is at cycle 7. Ready to be sent are strings: `axxxx`, `abc` and `da`. The shortest one is sent – `da`.

At the seventh cycle there is no incoming string. Ready to send are `axxxx`, `abc`, `bb` and `c`. The shortest one is sent – `c`.

At the next cycles there are no incoming strings. All strings in the system are ready - `axxxx`, `abc` and `bb`. They are sent one after another accordingly to their lengths.

NEW BULGARIAN UNIVERSITY
Department of Informatics
XXXIII National Student Programming Contest
May 8, 2021

Problem L. Common Substring

You have two strings p and q , which contain lower-case Latin letters. Find the length of the longest common substring, which is longer than a given length k .

Input.

For each test, the standard input contains the strings p and q and the number k .

Limitations.

The number k is greater than one-tenth of the length of the longer string and less than or equal to the length of the shorter string.

The lengths of the strings is in the interval $[2, 20000]$.

Output.

For each test, output the length of the common substring. If it is less than k , output 0.

Example:

<i>Input.</i>	<i>Output.</i>
abcd ab 2	2
qwerty eeertyyy 3	4
xz hey 2	0
xyzzz xxxyz 4	0

Example explanation:

The common substring `ab` is with length 2.

The common substring `erty` is with length 4.

No common substring.

The common substring `xyz` is with length $3 < 4$.