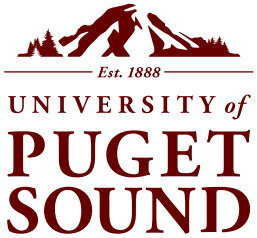
Pacific Northwest Region Programming Contest Division 2



November 5th, 2016

Reminders

- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Platform is as follows:

```
Ubuntu 16.04.1 LTS x86_64
geany
Java version OpenJDK 1.8.0_91
C/C++ gcc version 5.4.0
Eclipse 4.6 with CDT 9.0.1
Python 2.7.10 (IDE support with PyPy 5.1.2)
Python 3.5.2 (syntax highlighting editor support)
Pycharm 2016.2.3
```

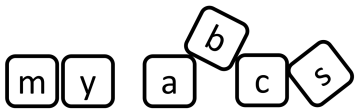
- Compiler options are as follows:

```
g++ -g -O2 -std=gnu++14 -static $*
gcc -g -O2 -std=gnu99 -static $* -lm
javac -encoding UTF-8 $*
java -client -Xss8m -Xmx1024m $*
python $*
mcs $*
mono $*
```

- Python may not have sufficient performance for many of the problems; use it at your discretion.

Problem M — limit 1 second

Alphabet



A string of lowercase letters is called *alphabetical* if deleting zero or more of its letters can result in the *alphabet string* "`abcdefghijklmnopqrstuvwxyz`".

Given a string s, determine the minimum number of letters to insert anywhere in the string to make it alphabetical.

Input

The input consists of a single line containing the string s ($1 \leq |s| \leq 50$).

It is guaranteed that s consists of lowercase ASCII letters '`a`' to '`z`' only.

Output

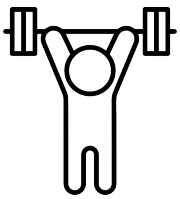
Print, on a single line, a single integer indicating the minimum number of letters that must be inserted in order to make the string s alphabetical.

Sample Input	**Sample Output**
`xyzabcdefghijklmnopqrstuvw`	`3`

Sample Input	**Sample Output**
`aiemckgobjfndlhp`	`20`

Problem N — limit 1 second

Barbells



Your local gym has n barbells and m plates. In order to prepare a weight for lifting, you must choose a single barbell, which has two sides. You then load each side with a (possibly empty) set of plates. For safety reasons, the plates on each side must sum to the same weight. What weights are available for lifting?

For example, suppose that there are two barbells weighing 100 and 110 grams, and five plates weighting 1, 4, 5, 5, and 6 grams, respectively. Then, there are six possible weights available for lifting. The table below shows one way to attain the different weights:

Barbell	Left side	Right side	Total
100	0	0	100
100	5	5	110
100	$1+5$	6	112
110	5	5	120
110	$1+5$	6	122
110	$5+5$	$4+6$	130

Input

The first line of input contains the space-separated integers n and m ($1 \leq n, m \leq 14$).

The second line of input contains n space-separated integers $b_1, \ldots, b_n$ ($1 \leq b_i \leq 10^8$), denoting the weights of the barbells in grams.

The third line of input contains m space-separated integers $p_1, \ldots, p_m$ ($1 \leq p_i \leq 10^8$), denoting the weights of the plates in grams.

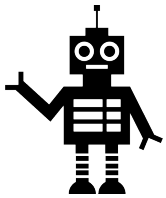
Output

Print a sorted list of all possible distinct weights in grams, one per line.

Sample Input	Sample Output
2 5 100 110 5 5 1 4 6	100 110 112 120 122 130

Problem O — limit 1 second

Buggy Robot



You are trying to program a robot to navigate through a 2-dimensional maze and find the exit.

The maze can be represented as a grid with n rows and m columns. Some grid cells have obstacles that the robot cannot pass. The other cells are empty, which the robot can freely pass. Exactly one of the empty cells in the grid is marked as the exit, and the robot will exit the maze immediately once it reaches there.

You can program the robot by sending it a *command string*. A command string consists of characters '`L`', '`U`', '`R`', '`D`', corresponding to the directions left, up, right, down, respectively. The robot will then start executing the commands, by moving to an adjacent cell in the directions specified by the command string. If the robot would run into an obstacle or off the edge of the grid, it will ignore the command, but it will continue on to remaining commands. The robot will also ignore all commands after reaching the exit cell.

Your friend sent you a draft of a command string, but you quickly realize that the command string will not necessarily take the robot to the exit. You would like to fix the string so that the robot will reach the exit square. In one second, you can delete an arbitrary character, or add an arbitrary character at an arbitrary position. Find how quickly you can fix your friend's command string.

You do not care how long it takes the robot to find the exit, but only how long it takes to repair the command string.

Input

The first line of input contains the two integers n and m ($1 \leq n, m \leq 50$).

Each of the next n lines contains m characters, describing the corresponding row of the grid. Empty cells are denoted as '`.`', the robot's initial position is denoted as '`R`', obstacles are denoted as '`#`', and the exit is denoted as '`E`'.

The next and final line of input contains your friend's command string, consisting of between 1 and 50 characters, inclusive.

It is guaranteed that the grid contains exactly one '`R`' and one '`E`', and that there is always a path from '`R`' to '`E`'.

Output

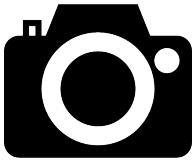
Print, on a single line, a single integer indicating the minimum amount of time to fix the program.

Sample Input	**Sample Output**
3 3 R.. .#. ..E LRDD	1

Sample Input	**Sample Output**
2 4 R.#. #..E RRUUDDRRUUUU	0

Problem P — limit 1 second

Cameras



Your street has n houses, conveniently numbered from 1 to n. Out of these n houses, k of them have security cameras installed. Mindful of gaps in coverage, the Neighborhood Watch would like to ensure that every set of r consecutive houses has at least two different houses with cameras. What is the minimum number of additional cameras necessary to achieve this?

Input

The first line of input contains three integers, n ($2 \leq n \leq 100{,}000$), k ($0 \leq k \leq n$), and r ($2 \leq r \leq n$).

The next k lines of input contain the distinct locations of the existing cameras.

Output

Print, on a single line, a single integer indicating the minimum number of cameras that need to be added.

Sample Input	Sample Output
15 5 4 2 5 7 10 13	3

Problem Q — limit 1 second

Contest Score

PC^2 SCOREBOARD			
Rank	Name	Solved	Points
1	RobotU	BuggyBot	42
2	MonstersInc	BuggyBot	43

You are participating in the Association for Computing Machinery's Intercollegiate Programming Competition (ACM ICPC). You must complete a set of n problems. Since you are an experienced problem solver, you can read a problem and accurately estimate how long it will take to solve it, in a negligible amount of time. But you can only keep the details from k problems straight at any given time.

Let t_i be the time it will take to solve the ith problem. Your strategy for the contest is as follows:

1. Read the first k problems.
2. Choose a problem that you have read that will take the shortest time to solve (if there are ties, choose any of them arbitrarily).
3. Solve the problem, and read the next unread problem (if there is any).
4. If there are still unsolved problems, go back to step 2.

You want to calculate your total penalty time for the contest. Your penalty time for the contest is defined by the sum of submission times for all the problems. Given the ordering of problems in the contest, compute the penalty time of your strategy.

Input

The first line of input contains two space-separated integers n and k ($1 \leq k \leq n \leq 300$).

The ith line of the next n lines contains a single integer t_i ($1 \leq t_i \leq 1{,}000{,}000$).

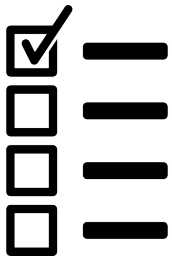
Output

Print, on a single line, a single integer repesenting the total penalty time.

Sample Input	Sample Output
4 3 1 3 2 1	14

Problem R — limit 1 second

Equality



You are grading an arithmetic quiz. The quiz asks a student for the sum of the numbers. Determine if the student taking the quiz got the question correct.

Input

The first and the only line of input contains a string of the form:

`a + b = c`

It is guaranteed that `a`, `b`, and `c` are single-digit positive integers. The input line will have exactly 9 characters, formatted exactly as shown, with a single space separating each number and arithmetic operator.

Output

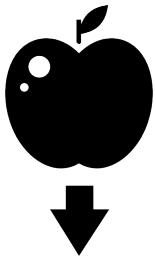
Print, on a single line, `YES` if the sum is correct; otherwise, print `NO`.

Sample Input	**Sample Output**
`1 + 2 = 3`	`YES`

Sample Input	**Sample Output**
`2 + 2 = 5`	`NO`

Problem S — limit 1 second

Gravity



You would like to implement a simple simulation of gravity on a 2-dimensional grid, consisting of n rows and m columns.

Some grid cells may contain obstacles, some may contain a single apple, and all others are empty.

The following rules are followed until no further changes are possible:

- The obstacles do not move.
- Whenever there is an empty cell immediately below an apple, the apple moves into the empty cell.

Find the final configuration of the grid after all apples have settled.

Input

The first line of input contains two space-separated integers n and m ($1 \leq n, m \leq 50$).

Each of the next n lines contains m characters, describing the board, from top to bottom. Obstacles are denoted as '`#`', apples are denoted as '`o`', and empty cells are denoted as '`.`'.

Output

Print, on n lines, the final configuration of the grid after executing the rules stated above.

Sample Input	**Sample Output**
3 3 ooo #.. ..#	o.. #.o .o#

Sample Input	Sample Output
4 2 oo oo o. ..	.. o. oo oo

Problem T — limit 1 second

Islands

You are mapping a faraway planet using a satellite.

Your satellite has captured an image of the planet's surface. The photographed section can be modeled as a grid. Each grid cell is either land, water, or covered by clouds. Clouds mean that the surface could either be land or water, but we can't tell.

An island is a set of connected land cells. Two cells are considered connected if they share an edge.

Given the image, determine the minimum number of islands that is consistent with the given information.

Input

The first line of input contains two space-separated integers n and m ($1 \leq n, m \leq 50$).

Each of the next n lines contains m characters, describing the satellite image. Land cells are denoted by '`L`', water cells are denoted by '`W`', and cells covered by clouds are denoted by '`C`'.

Output

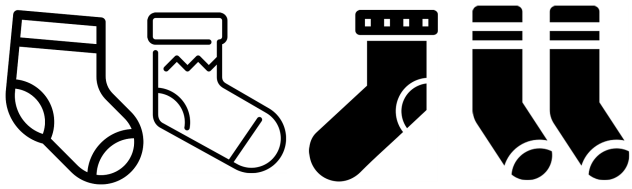
Print, on a single line, a single integer indicating the minimum number of islands that is consistent with the given grid.

Sample Input	**Sample Output**
4 5 CCCCC CCCCC CCCCC CCCCC	0

Sample Input	Sample Output
3 2 LW CC WL	1

Problem U — limit 1 second

Mismatched Socks



Fred likes to wear mismatched socks. This sometimes means he has to plan ahead.

Suppose his sock drawer has 1 red, 1 blue, and 2 green socks. If he wears the red with the blue, he is stuck with matching green socks the next day. Given the contents of his sock drawer, how many pairs of mismatched socks can he put together?

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), the number of different colors of socks in Fred's drawer.

The ith of the next n lines contains a single integer k_i ($1 \leq k_i \leq 10^9$), the number of socks of the ith color.

Output

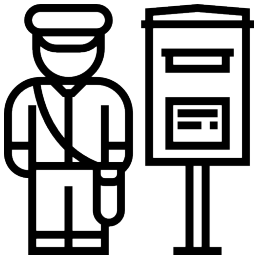
Print, on a single line, the maximum number of mismatched pairs of socks that Fred can make with the contents of his sock drawer.

Sample Input	Sample Output
3 1 2 1	2

Sample Input	Sample Output
5 1 2 1 10 3	7

Problem V — limit 1 second

Postman



A postman delivers letters to his neighbors in a one-dimensional world.

The post office, which contains all of the letters to begin with, is located at $x = 0$, and there are n houses to which the postman needs to deliver the letters. House i is located at position x_i, and there are m_i letters that need to be delivered to this location. But the postman can only carry k letters at once.

The postman must start at the post office, pick up some number of letters less than or equal to his carrying capacity, and then travel to some of the houses dropping off letters. He must then return to the post office, repeating this process until all letters are delivered. At the end he must return to the post office.

The postman can travel one unit of distance in one unit of time.

What is the minimum amount of time it will take the postman to start at the post office, deliver all the letters, and return to the post office?

Input

The first line of input contains two space-separated integers n ($1 \leq n \leq 1{,}000$) and k ($1 \leq k \leq 10^7$).

Each of the next n lines contains two space-separated integers x_i ($|x_i| \leq 10^7$) and m_i ($1 \leq m_i \leq 10^7$).

Output

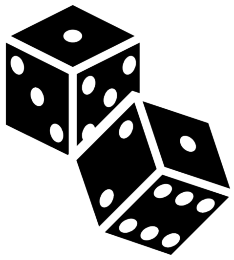
Print, on a single line, the minimum amount of time it will take to complete the mail delivery route.

Sample Input	Sample Output
4 10 -7 5 -2 3 5 7 9 5	42

Sample Input	Sample Output
7 1 9400000 10000000 9500000 10000000 9600000 10000000 9700000 10000000 9800000 10000000 9900000 10000000 10000000 10000000	1358000000000000

Problem W — limit 1 second

Six Sides



Two players play a simple game. Each brings their own six-sided die with specified values on the six faces. Each die is fair; that is, when it is thrown, each of its six faces is equally likely to come up on top.

The first player throws the first die and the second throws the second die. If the values shown on the top of the dice differ, the player with the higher value wins. If the values are the same, both players throw the dice again.

Given two dice with specific values, what is the probability that the first player wins?

Input

The first line of input contains six space-separated integers, representing the values written on the first player's die.

The second line of input contains the values on the second player's die in the same format.

It is guaranteed that all the values are between 1 and 6, inclusive.

Output

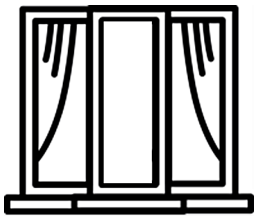
Print, on a single line, a floating-point value representing the probability that the first player wins, rounded and displayed to exactly five decimal places. The value should be printed with one digit before the decimal point and five digits after the decimal point. The sixth digit after the decimal point of the exact answer will never be 4 or 5 (eliminating complex rounding considerations).

Sample Input	Sample Output
1 2 3 4 5 6 1 2 3 4 5 6	0.50000

Sample Input	Sample Output
4 4 4 4 1 1 3 3 3 3 3 3	0.66667

Problem X — limit 1 second

Three Square



Inspired by a Mondrian painting you saw, you want to make a three-pane window out of three colored rectangles of glass. You are given the sizes of the three panes of glass. Can you arrange them into a square? You may rotate the panes, but the panes may not overlap.

Input

The input consists of three lines. Each of the three lines gives the length and height of a pane of glass.

It is guaranteed that the lengths and heights are integers between 1 and 100, inclusive.

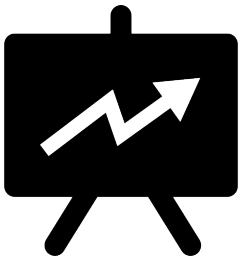
Output

Print, on a single line, `YES` if the panes of glass can be combined to form a square; otherwise, print `NO`.

Sample Input	Sample Output
8 2 1 6 7 6	YES

Problem Y — limit 1 second

Zigzag



Your Ph.D. thesis on properties of integer sequences is coming along nicely. Each chapter is on a different type of sequence. The first covers arithmetic sequences. Subsequently you cover binomial sequences, computable sequences, decreasing sequences, and so on. You have one more chapter to write, on zigzagging sequences.

A sequence is *zigzagging* if adjacent elements alternate between strictly increasing and strictly decreasing. The first pair of numbers can be either strictly increasing or strictly decreasing.

For a given sequence, find the length of its longest zigzagging subsequence.

Input

The first line of input contains a single integer n ($1 \leq n \leq 50$), the length of the sequence.

The second line contains n space-separated integers, describing the sequence.

Every number in the sequence is guaranteed to be between 1 and 50, inclusive.

Output

Print, on a single line, the length of a longest zigzagging subsequence of the input sequence.

Sample Input	**Sample Output**
5 2 1 3 4 2	4

Sample Input	Sample Output
10 1 1 1 1 1 1 1 1 1 1	1