A. Chat room

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya has recently learned to type and log on to the Internet. He immediately entered a chat room and decided to say hello to everybody. Vasya typed the word *s*. It is considered that Vasya managed to say hello if several letters can be deleted from the typed word so that it resulted in the word "hello". For example, if Vasya types the word "ahhellllloou", it will be considered that he said hello, and if he types "hlelo", it will be considered that Vasya got misunderstood and he didn't manage to say hello. Determine whether Vasya managed to say hello by the given word *s*.

Input

The first and only line contains the word *s*, which Vasya typed. This word consisits of small Latin letters, its length is no less that 1 and no more than 100 letters.

Output

If Vasya managed to say hello, print "YES", otherwise print "NO".

Sample test(s)

input
ahhellllloou
output
YES

input
hlelo
output
NO

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, than value 800 ms will be displayed and used to determine the verdict.

Codeforces Beta Round #54 (Div. 2)

Finished

Practice

→ Submit?

Language: GNU C++0x 4

Choose file: Browse... No file selected.

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Contest materials

- Announcement
- Tutorial

Codeforces (c) Copyright 2010-2014 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: 2014-07-28 18:19:32 (c4).

A. Cinema Line

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The new "Die Hard" movie has just been released! There are n people at the cinema box office standing in a huge line. Each of them has a single 100, 50 or 25 ruble bill. A "Die Hard" ticket costs 25 rubles. Can the booking clerk sell a ticket to each person and give the change if he initially has no money and sells the tickets strictly in the order people follow in the line?

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of people in the line. The next line contains n integers, each of them equals 25, 50 or 100 — the values of the bills the people have. The numbers are given in the order from the beginning of the line (at the box office) to the end of the line.

Output

Print "YES" (without the quotes) if the booking clerk can sell a ticket to each person and give the change. Otherwise print "NO".

Sample test(s)

input
4 25 25 50 50
output
YES

input
2 25 100
output
NO

input
4 50 50 25 25
output
NO

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, than value 800 ms will be displayed and used to determine the verdict.

Codeforces Round #202 (Div. 2)

Finished

Practice

→ Submit?

Language: GNU C++0x 4

Choose file: Browse... No file selected.

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Contest materials

- Announcement
- Tutorial

Codeforces (c) Copyright 2010-2014 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: 2014-07-28 18:19:32 (c4).

B. Levko and Array

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Levko has an array that consists of integers: $a_1, a_2, \ldots, a_n$. But he doesn't like this array at all.

Levko thinks that the beauty of the array a directly depends on value $c(a)$, which can be calculated by the formula:

$$c(a) = \max_{1 \le i \le n-1} |a_{i+1} - a_i|, n > 1; c(a) = 0, 0 \le n \le 1.$$

The less value $c(a)$ is, the more beautiful the array is.
It's time to change the world and Levko is going to change his array for the better. To be exact, Levko wants to change the values of at most k array elements (it is allowed to replace the values by any integers). Of course, the changes should make the array as beautiful as possible.

Help Levko and calculate what minimum number $c(a)$ he can reach.

Input

The first line contains two integers n and k ($1 \le k \le n \le 2000$). The second line contains space-separated integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

Output

A single number — the minimum value of $c(a)$ Levko can get.

Sample test(s)

input
5 2 4 7 4 7 4

output
0

input
3 1 -100 0 100

output
100

input
6 3 1 2 3 7 8 9

output
1

Note

In the first sample Levko can change the second and fourth elements and get array: 4, 4, 4, 4, 4.

In the third sample he can get array: 1, 2, 3, 4, 5, 6.

Codeforces Round #210 (Div. 1)

Finished

Practice

→ Submit?

Language: GNU C++0x 4

Choose file: Browse... No file selected.

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Contest materials

- Announcement
- Tutorial

Codeforces (c) Copyright 2010-2014 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: 2014-07-28 18:19:32 (c4).

A. Ciel and Robot

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel has a robot on a 2D plane. Initially it is located in (0, 0). Fox Ciel code a command to it. The command was represented by string s. Each character of s is one move operation. There are four move operations at all:

- 'U': go up, (x, y) → (x, y+1);
- 'D': go down, (x, y) → (x, y-1);
- 'L': go left, (x, y) → (x-1, y);
- 'R': go right, (x, y) → (x+1, y).

The robot will do the operations in s from left to right, and repeat it infinite times. Help Fox Ciel to determine if after some steps the robot will located in (a, b).

Input

The first line contains two integers a and b, ($-10^9 \le a, b \le 10^9$). The second line contains a string s ($1 \le |s| \le 100$, s only contains characters 'U', 'D', 'L', 'R') — the command.

Output

Print "Yes" if the robot will be located at (a, b), and "No" otherwise.

Sample test(s)

input
2 2 RU
output
Yes

input
1 2 RU
output
No

input
-1 1000000000 LRRLU
output
Yes

input
0 0 D
output
Yes

Note

In the first and second test case, command string is "RU", so the robot will go right, then go up, then right, and then up and so on.

The locations of its moves are (0, 0) → (1, 0) → (1, 1) → (2, 1) → (2, 2) → ...

So it can reach (2, 2) but not (1, 2).

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, than value 800 ms will be displayed and used to determine the verdict.

Codeforces Round #190 (Div. 1)

Finished

Practice

→ Submit?

Language: GNU C++0x 4

Choose file: Browse... No file selected.

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Contest materials

- Tutorial

SPOJ Problem Set (classical)

0
g+1

1674. The Explosion

Problem code: EXPLOSN

The day of 6.XII.2003 in Megabyteland began calm and quietly as any other day. Some people went to work, some - to school, some - to store to buy food. Drivers were traditionally stucked in traffic jams, drinking coffee and reading morning newspaper. Suddenly the regularity of this day was disturbed by huge explosion."They blew up the embassy of Bajtocja!!!" - somebody cried. Everybody began to run away in panic.

Police works pretty good in Megabyteland and first radiocars appeared near the embassy only few seconds after the explosion. All the people near the embassy were detained. Some of these people are the organizers of the explosion, but the others could by just occasional witnesses. During the testification each person named exactly one perpetrator. It is known, that if a man is not a perpetrator, than he always says the truth (he haven't a reason to lie, have he?). However, perpetrators want to make the work of the police more difficult, so a perpetrator can name any person during the testification (even himself).

The policemen are in the very hard situation. They should arrest some group of potential perpetrators, but it is difficult to determine who is guilty and who is not from the data they have. There exists many groups of potential perpetrators, that don't contradict to any of the testimonies. The policemen want to arrest as small innocent people as possible. So they would like to choose the group with minimal number of people.

Write a program that, given the number of detained people and their testimonies, will determine the number of people in the smallest group of potential perpetrators, that don't contradict to the testimonies.

Input

The first line of the input contains a single integer T, the number of testcases (1<=T<=10).

First line of each testcase contains integer number N (2 <= N <= 100000), equal to the number of detained people (the people are numbered from 1 to N). The i-th of the following N lines contain one integer number Pi (1 <= Pi <= N). Here Pi is the man whom i-th man testified to be guilty.

Output

The output should consist of T lines, containing one integer number for each testcase - the number of people in the smallest group of potential perpetrators, that don't contradict to the testimonies.

Example

Input:
```
1
3
2
3
1
```

Output:
```
2
```


© Spoj.com. All Rights Reserved. Spoj uses Sphere Engine™ © by Sphere Research Labs.

C. Games with Rectangle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this task Anna and Maria play the following game. Initially they have a checkered piece of paper with a painted $n \times m$ rectangle (only the border, no filling). Anna and Maria move in turns and Anna starts. During each move one should paint inside the last-painted rectangle a new lesser rectangle (along the grid lines). The new rectangle should have no common points with the previous one. Note that when we paint a rectangle, we always paint only the border, the rectangles aren't filled.

Nobody wins the game — Anna and Maria simply play until they have done k moves in total. Count the number of different ways to play this game.

Input

The first and only line contains three integers: n, m, k ($1 \le n, m, k \le 1000$).

Output

Print the single number — the number of the ways to play the game. As this number can be very big, print the value modulo 1000000007 ($10^9 + 7$).

Sample test(s)

input
3 3 1
output
1

input
4 4 1
output
9

input
6 7 2
output
75

Note

Two ways to play the game are considered different if the final pictures are different. In other words, if one way contains a rectangle that is not contained in the other way.

In the first sample Anna, who performs her first and only move, has only one possible action plan — insert a 1×1 square inside the given 3×3 square.

In the second sample Anna has as much as 9 variants: 4 ways to paint a 1×1 square, 2 ways to insert a 1×2 rectangle vertically, 2 more ways to insert it horizontally and one more way is to insert a 2×2 square.

Codeforces Beta Round #94 (Div. 1 Only)

Finished

Practice

→ Submit?

Language: GNU C++0x 4

Choose file: Browse... No file selected.

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

C. Volleyball

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya loves volleyball very much. One day he was running late for a volleyball match. Petya hasn't bought his own car yet, that's why he had to take a taxi. The city has n junctions, some of which are connected by two-way roads. The length of each road is defined by some positive integer number of meters; the roads can have different lengths.

Initially each junction has exactly one taxi standing there. The taxi driver from the i-th junction agrees to drive Petya (perhaps through several intermediate junctions) to some other junction if the travel distance is not more than t_i meters. Also, the cost of the ride doesn't depend on the distance and is equal to c_i bourles. Taxis can't stop in the middle of a road. **Each taxi can be used no more than once. Petya can catch taxi only in the junction, where it stands initially.**

At the moment Petya is located on the junction x and the volleyball stadium is on the junction y. Determine the minimum amount of money Petya will need to drive to the stadium.

Input

The first line contains two integers n and m ($1 \le n \le 1000, 0 \le m \le 1000$). They are the number of junctions and roads in the city correspondingly. The junctions are numbered from 1 to n, inclusive. The next line contains two integers x and y ($1 \le x, y \le n$). They are the numbers of the initial and final junctions correspondingly. Next m lines contain the roads' description. Each road is described by a group of three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n, 1 \le w_i \le 10^9$) — they are the numbers of the junctions connected by the road and the length of the road, correspondingly. The next n lines contain n pairs of integers t_i and c_i ($1 \le t_i, c_i \le 10^9$), which describe the taxi driver that waits at the i-th junction — the maximum distance he can drive and the drive's cost. The road can't connect the junction with itself, but between a pair of junctions there can be more than one road. All consecutive numbers in each line are separated by exactly one space character.

Output

If taxis can't drive Petya to the destination point, print "-1" (without the quotes). Otherwise, print the drive's minimum cost.

Please do not use the %lld specificator to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the %I64d specificator.

Sample test(s)

input
4 4 1 3 1 2 3 1 4 1 2 4 1 2 3 5 2 7 7 2 1 2 7 7
output
9

Note

An optimal way — ride from the junction 1 to 2 (via junction 4), then from 2 to 3. It costs 7+2=9 bourles.

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, than value 800 ms will be displayed and used to determine the verdict.

Codeforces Beta Round #77 (Div. 1 Only)
Finished
Practice

→ Submit?

Language: GNU C++0x 4
Choose file: Browse... No file selected.
Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.
Submit

C. Looking for Order

time limit per test: 7.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Girl Lena likes it when everything is in order, and looks for order everywhere. Once she was getting ready for the University and noticed that the room was in a mess — all the objects from her handbag were thrown about the room. Of course, she wanted to put them back into her handbag. The problem is that the girl cannot carry more than two objects at a time, and cannot move the handbag. Also, if he has taken an object, she cannot put it anywhere except her handbag — her inherent sense of order does not let her do so.

You are given the coordinates of the handbag and the coordinates of the objects in some Cartesian coordinate system. It is known that the girl covers the distance between any two objects in the time equal to the squared length of the segment between the points of the objects. It is also known that initially the coordinates of the girl and the handbag are the same. You are asked to find such an order of actions, that the girl can put all the objects back into her handbag in a minimum time period.

Input

The first line of the input file contains the handbag's coordinates x_s, y_s. The second line contains number n ($1 \le n \le 24$) — the amount of objects the girl has. The following n lines contain the objects' coordinates. All the coordinates do not exceed 100 in absolute value. All the given positions are different. All the numbers are integer.

Output

In the first line output the only number — the minimum time the girl needs to put the objects into her handbag.

In the second line output the possible optimum way for Lena. Each object in the input is described by its index number (from 1 to n), the handbag's point is described by number 0. The path should start and end in the handbag's point. If there are several optimal paths, print any of them.

Sample test(s)

input
0 0 2 1 1 -1 1
output
8 0 1 2 0

input
1 1 3 4 3 3 4 0 0
output
32 0 1 2 0 3 0

SPOJ Problem Set (classical)

2

g+1

740. Treats for the Cows

Problem code: TRT

FJ has purchased N (1 <= N <= 2000) yummy treats for the cows who get money for giving vast amounts of milk. FJ sells one treat per day and wants to maximize the money he receives over a given period time. The treats are interesting for many reasons:

- The treats are numbered 1..N and stored sequentially in single file in a long box that is open at both ends. On any day, FJ can retrieve one treat from either end of his stash of treats.
- Like fine wines and delicious cheeses, the treats improve with age and command greater prices.
- The treats are not uniform: some are better and have higher intrinsic value. Treat i has value v(i) (1 <= v(i) <= 1000).
- Cows pay more for treats that have aged longer: a cow will pay v(i)*a for a treat of age a.

Given the values v(i) of each of the treats lined up in order of the index i in their box, what is the greatest value FJ can receive for them if he orders their sale optimally?

The first treat is sold on day 1 and has age a=1. Each subsequent day increases the age by 1.

Input

Line 1: A single integer, N

Lines 2..N+1: Line i+1 contains the value of treat v(i)

Output

The maximum revenue FJ can achieve by selling the treats

Example

Input:
```
5
1
3
1
5
2
```

Output:
```
43
```


© Spoj.com. All Rights Reserved. Spoj uses Sphere Engine™ © by Sphere Research Labs.

ZOJ Problem Set - 1028

Flip and Shift

Time Limit: 2 Seconds Memory Limit: 65536 KB

This puzzle consists of a random sequence of m black disks and n white disks on an oval-shaped track, with a turnstile capable of flipping (i.e., reversing) three consecutive disks. In Figure 1, there are 8 black disks and 10 white disks on the track. You may spin the turnstile to flip the three disks in it or shift one position clockwise for each of the disks on the track (Figure 1).

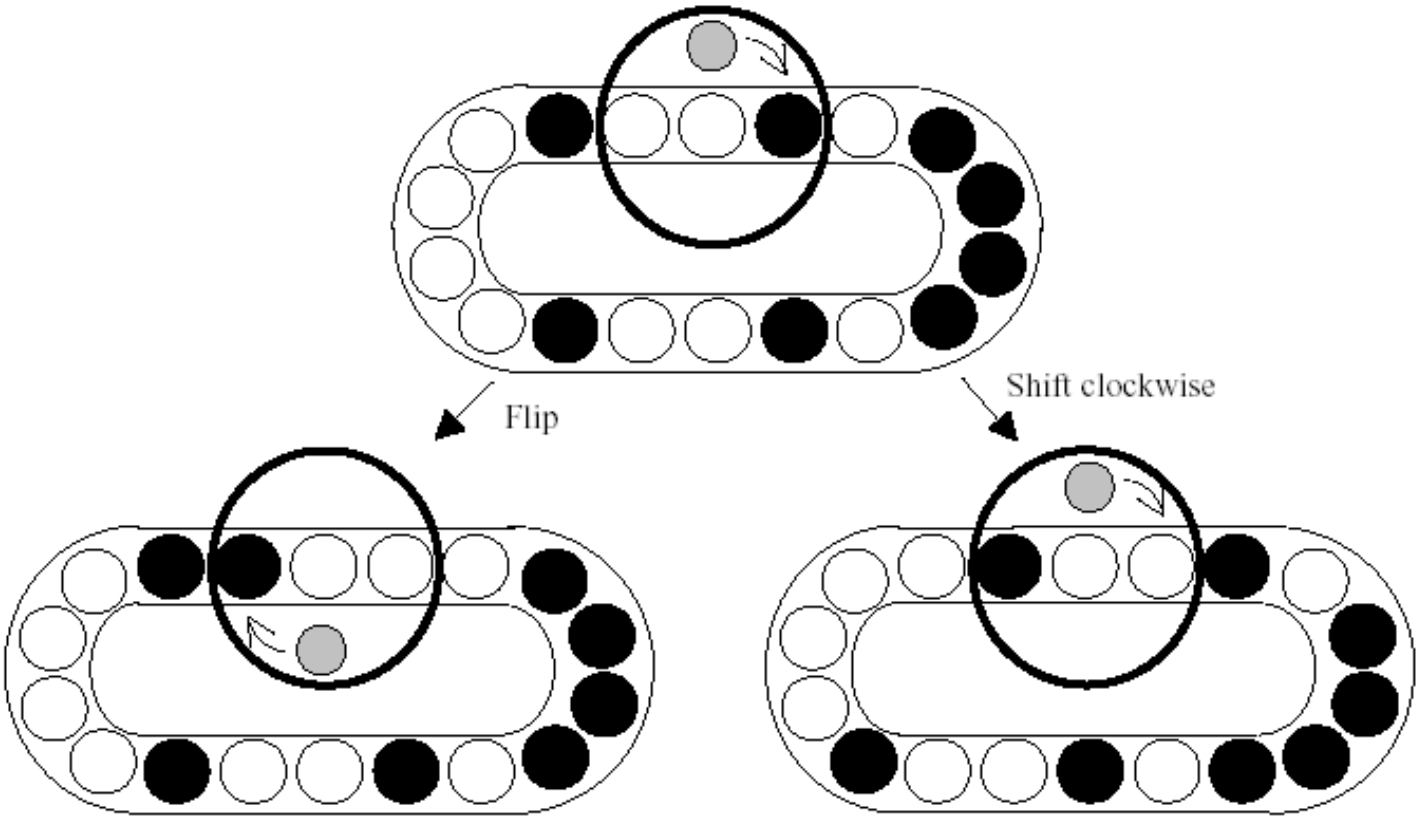

Figure 1. A flip and a shift

The goal of this puzzle is to gather the disks of the same color in adjacent positions using flips and shifts. (Figure 2)

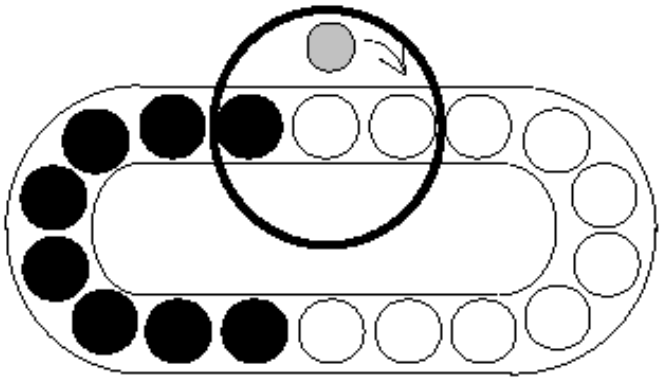

Figure 2. A goal sequence

You are to write a program which decides whether a given sequence can reach a goal or not. If a goal is reachable, then write a message "YES"; otherwise, write a message "NO".

Input

The input consists of T test cases. The number of test cases) (T is given in the first line of the input. Each of the next T lines gives a test case. A test case consists of an integer, representing the sum of m and n, and a sequence of m+n 0s and 1s, representing an initial sequence. A 0 denotes a white disk and a 1 denotes a black disk. The sum of m and n is at least 10 and does not exceed 30. There is a space between numbers.

Output

The output should print either "YES" or "NO" for each test case, one per line.

Sample Input

```
2
18 0 0 1 0 1 1 1 1 0 1 0 0 1 0 0 0 0 1
14 1 1 0 0 1 1 1 0 0 1 1 0 1 0
```

Output for the Sample Input

```
YES
NO
```

Source: **Asia 2001, Taejon (South Korea)**

Submit Status