South African Computer Olympiad
1st Training Camp 2011
Training Camp Day 2

Overview

Author(s)	**Russ Cox**	**IOI 2005**	**Keegan Carruthers-Smith**	**BOI**
Problem	**events**	**polflag**	**bus**	**sound**
Source	events.c events.cpp	polflag.c polflag.cpp	bus.c bus.cpp	sound.c sound.cpp
Input file	stdin	stdin	stdin	stdin
Output file	stdout	stdout	stdout	stdout
Time limit	1 second	0.1 seconds	1 second	.21.2 second
Memory limit	64MiB	64MiB	64MiB	64MiB
Number of tests	10	20	10	10
Points per test	10	5	10	10
Detailed feedback	No	No	Yes	No
Total points	**100**	**100**	**100**	**100**

The maximum total score is 400 points.

County Fair Events

Russ Cox

Problem by Russ Cox from USACO Silver April 2005

Introduction

Farmer John has returned to the County Fair so he can attend the special events (concerts, rodeos, cooking shows, etc.). He wants to attend as many of the N special events as he possibly can.

He's rented a bicycle so he can speed from one event to the next in absolutely no time at all (0 time units to go from one event to the next!).

Task

Given a list of the events that FJ might wish to attend, with their start times ($1 \leq T \leq 100\,000$) and their durations ($1 \leq L \leq 100\,000$), determine the maximum number of events that FJ can attend. FJ never leaves an event early.

Example

Here is a graphic picture representing the schedule in the sample input

```
          11111111112
12345678901234567890
--------------------
111111  2222223333344
55555555  777777   666
```

FJ can do no better than to attend events 1, 2, 3, and 4.

Input (stdin)

The first line contains a single integer, N. The following N lines each contain two space-separated integers, T and L, that describe an event that FJ might attend.

Sample input

```
7
1 6
8 6
14 5
19 2
1 8
18 3
10 6
```

Output (stdout)

A single integer that is the maximum number of events FJ can attend.

Sample output

```
4
```

Constraints

- $1 \leq N \leq 10\,000$
- $1 \leq T \leq 100\,000$
- $1 \leq L \leq 100\,000$

Additionally, in 30% of the test cases n will not exceed 10.

Time limit

1 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Polish Flag

IOI 2005

Problem from IOI 2005 Sample Problems

Introduction

Three children are building Polish flag from square blocks. The flag will be a rectangle, $3n$ blocks wide and $2n$ blocks high, where n is a positive integer. It will consist of $3n^2$ white blocks and $3n^2$ red blocks. The children are going to lay blocks on a rectangle table. There are $6n^2$ slots on the table. The white blocks should occupy the top n rows, and the red blocks should occupy the bottom n rows. Rows are numbered from 1 to $2n$ from top to bottom. Columns are numbered from 1 to $3n$ from left to right.

The children are laying blocks in turns. In the first turn Lucy puts her block on the left edge at position $(1, l)$, Bob puts his block on the bottom edge at position $(b, 2n)$, Roy puts his block on the right edge at position $(3n, r)$, where $1 \leq l, r < 2n$, $1 < b < 3n$.

Every next turn they lay blocks as follows. The child can put a block in a given slot only if the slot is empty and the block to be put would be adjacent to one of the blocks put in the preceding turn. (Two blocks are adjacent if they have a common side.) In a given turn the child puts as many blocks as possible. Only one block can be put into a single slot. If two or more children want to put a block into the same slot in the same turn then the highest priority has Lucy, then Bob and the lowest priority has Roy.

Task

Before the children start building the flag they have to distribute blocks. Here is the problem. They don't know how many blocks of each color they need. Help the children and compute for each child the number of blocks of each color he/she will use while building the flag.

Example

Input (stdin)

The first and only line contains four integers n, l, b, r, separated by single spaces.

Sample input

```
2 2 3 1
```

Output (stdout)

Output should consist of a single line containing six integers separated by single spaces. The first and the second integer should be the number of white and red blocks respectively, which Lucy needs; the third and fourth number should be the number of white and red blocks respectively, which Bob needs; the fifth and sixth number should be the number of white and red blocks respectively, which Roy needs.

Sample output

```
7 3 0 8 5 1
```

Constraints

- $1 \leq n \leq 1\,000\,000\,000$
- $1 \leq l, r < 2n$
- $1 < b < 3n$.

Additionally, in 50% of test cases n will not exceed 100.

Time limit

0.1 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

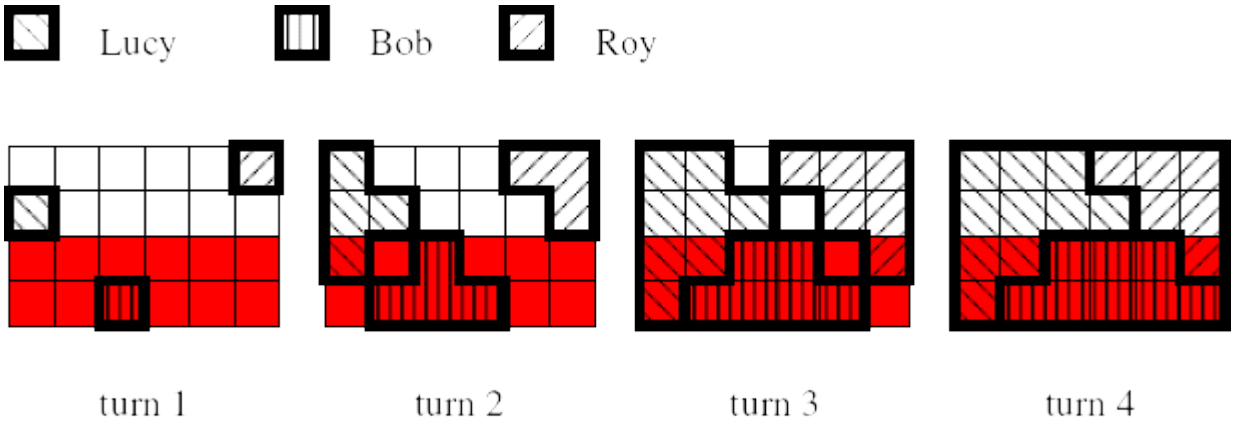

Figure 1: This figure represents the sample input.

Bus Routes

Keegan Carruthers-Smith

Problem by Keegan Carruthers-Smith

Introduction

The government wants to create more bus routes in South Africa. Every bus route has buses traveling in both directions on it. Unfortunately, it is not always possible to get from a city to every other city using only buses. So the government wants to create more bus routes, such that it is possible to travel between any 2 cities just using buses.

Task

Creating a bus route costs money, so your task is to work out the minimum cost needed such that there is a path between any 2 cities using only buses.

Example

In the example there already exists bus routes between the cities $4 \leftrightarrow 2$ and between cities $3 \leftrightarrow 5$. If we build the bus routes $2 \leftrightarrow 3$ and $1 \leftrightarrow 3$, we can reach any city using the layout. This has a cost of 150, which is not minimal. However if we build the bus routes $2 \leftrightarrow 1$ (of cost 2) and $3 \leftrightarrow 1$, we can reach any city. This has a cost of 52, which is minimal.

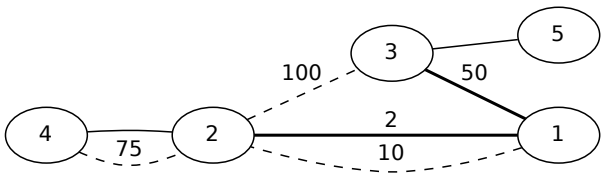

Figure 2: This represents the sample input. Non-bold solid lines represent bus routes that already exist. Bold solid lines represent the bus routes picked to minimize the cost of the task. Dashed lines represent other bus routes which where not picked.

Input (stdin)

The first line of the input contains 3 space-separated integers, N, C and M. There are N cities, numbered 1 to N. The next C lines each contain two space-separated integers a and b. This represents an already existing bus route between a and b. The next M lines each contain 3 space-separated a, b and c. This represents a bus route you can build between a and b with cost c.

Sample input

```
5 2 5
4 2
3 5
2 3 100
3 1 50
1 2 10
4 2 75
2 1 2
```

Output (stdout)

The output contains a single integer, the minimum cost needed such that there is a path between any two cities.

Sample Output

```
52
```

Constraints

- $1 \leq N \leq 1\,000$
- $0 \leq C, M \leq 10\,000$
- $1 \leq a, b \leq N$
- $1 \leq c \leq 1\,000$
- There is always a way to build bus routes such that there is a path between any 2 cities.

Time limit

1 second.

Detailed feedback

Detailed feedback is enabled for this problem.

Scoring

100% per test case for a correct answer. 0% otherwise.

The Sound of Silence

BOI

Problem from BOI 2007

Introduction

In digital recording, sound is described by a sequence of numbers representing the air pressure, measured at a rapid rate with a fixed time interval between successive measurements. Each value in the sequence is called a sample.

An important step in many voice-processing tasks is breaking the recorded sound into chunks of non-silence separated by silence. To avoid accidentally breaking the recording into too few or too many pieces, the silence is often defined as a sequence of M samples where the difference between the lowest and the highest value does not exceed a certain threshold C.

Task

Write a program to detect silence in a given recording of N samples according to the given parameter values M and C.

Example

In the example input there are only two times that the difference between the highest and lowest samples in the last 2 values is less than equal to 0. That is at position 2 ($[1, 1]$) and at position 6 ($[2, 2]$). Note that positions are 1-indexed (the first position is not at 0, but at 1).

Input (stdin)

The first line of input contains three space-separated integers: N, M and C. N is the number of samples in the recording. M is the required length of the silence. C is the maximal noise level allowed within silence.

The second line of the file contains N integers a_i, seperated by single spaces: the samples in the recording.

Sample input

```
7 2 0
0 1 1 2 3 2 2
```

Output (stdout)

The output lists all values of i such that

$$\max(a_i, a_{i+1}, \ldots, a_{i+M-1}) - \min(a_i, a_{i+1}, \ldots, a_{i+M-1}) \leq C$$

The values should be listen in increasing order, each on a separate line.

If there is no silence in the input file, write `NONE` on the first and only line of the output file.

Sample output

```
2
6
```

Constraints

- $1 \leq N \leq 1\,000\,000$
- $1 \leq M \leq 10\,000$
- $0 \leq C \leq 10\,000$
- $0 \leq a_i \leq 1\,000\,000$ for $1 \leq i \leq N$

Time limit

.21.2 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.