0 SimpleSum

0.1 Problem

Given a sequence of n integers $a_1, \ldots, a_n$, $1 \leq n \leq 1000$, $0 \leq a_i \leq 100$, compute the sum $s = \sum_{i=1}^{n} a_i$.

0.2 Input

The first line contains a single integer n. n lines follow, with the ith line containing a single integer representing a_i.

0.3 Output

Output the sum s followed by a newline character.

0.4 Sample Data

Input	Output
3 1 2 42	45

1 The 3n + 1 problem

1.1 Problem

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g. NP, Unsolvable). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

1. input n
2. print n
3. if $n = 1$ then STOP
4. if n is odd, then $n := 3n + 1$
5. else $n := n/2$
6. goto 2

Given the input 22, the following sequence of numbers will be printed:

$$22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1$$

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers in $\{1, \ldots, 1000000\}$ (and, in fact, for many more numbers than this).

Given an input n, it is possible to determine the length of the sequence of printed numbers (including the 1). For a given n this is called the Collatz-length of n. In the example above, the Collatz-length of 22 is 16. For two given numbers i and j you are to determine the maximum Collatz-length of numbers in $[i, j]$.

1.2 Input

The input consists of a pair of integers i and j ($1 \leq i \leq j < 1000$).

1.3 Output

Output the maximum Collatz length for starting values in $\{i, i + 1, \ldots, j - 1, j\}$, followed by a newline character.

1.4 Sample Data

Input	Output
1 10	20
100 200	125
201 210	89
900 999	174

2 Chocolate Bar

2.1 Problem

Assume that you have a chocolate bar consisting, as usual, of $w \times h$ small squares arranged in a rectangular pattern. Your task is to split the bar into the squares (pieces of size 1×1) with a minimum number of breaks.

A break is a straight line cut along one of the lines between the squares. You can only break one connected piece of chocolate at a time and cannot break the squares. For example, stacking two pieces and breaking them in one go counts as two breaks.

How many breaks do you need at least?

2.2 Input

The first line of the input contains an integer t, the number of testcases ($1 \leq t \leq 10^5$). The following t lines contain two integers w and h each ($1 \leq w, h \leq 2^{13}$).

2.3 Output

Output t lines, with line i containing the smallest number of breaks for the i-th testcase. Each line (including the last) should be terminated by a single newline character.

2.4 Sample Data

Input	Output
3 6 5 1 100 6423 4245	29 99 27265634

3 Professor Snørbørden's Slides

3.1 Problem

Marcus is a smart and diligent computer science student but the lectures of his Professor Snørbørden are not exactly the most thrilling moments of his studies. In other words: Professor Snørbørden is exceptionally boring. Marcus can hardly stay awake when listening to the slow, monotonous words of the Professor, and the only joy left for him is that of pointing out the mistakes on the slides to Professor Snørbørden. Since the slides were designed by Professor Snørbørden himself, the errors are blatantly obvious and can easily be spotted by a clever student such as Marcus when flipping through them on the day before the lecture.

Though Marcus wants to point out as many mistakes as possible to the professor for his own amusement, the sedative power of Snørbørden's voice is so overwhelming, that Marcus cannot help but get some sleep during the lecture. Having seen all the slides in advance he can predict the boredom and therefore the amount of sleep that he needs. The amount of sleep is measured in slides, which is an accurate way to measure time in Professor Snørbørden's lecture. Marcus has control over when he starts sleeping, but being an experienced sleeper, Marcus also knows that he has to sleep in one long block rather than fragmenting his sleep throughout the lecture in order for the sleep to be revitalising. Sadly this restriction may cause Marcus to miss some of the flawed slides and therefore deny the auditorium the fun of yet another instance of a confused Professor Snørbørden which is the only entertainment there is.

Though Marcus is - as mentioned above - pretty clever, his problem solving skills deteriorated recently (probably due to being exposed to Professor Snørbørden). Marcus therefore needs your help to determine when to start his sleep in order to miss as few flawed slides as possible.

3.2 Input

The input contains two integers s and n ($1 \leq s \leq n \leq 10^5$), where s is the amount of sleep that Marcus needs (measured in slides) and n the number of slides in the lecture. The second line contains n numbers $b_0, b_1, ...b_{n-1}$, each of which is either 0 or 1. b_i is 0 if the ith slide is correct, or 1 if the ith slide is flawed.

3.3 Output

You should output the number of the slide before which Marcus should start sleeping in order to miss as few flawed slides as possible, followed by a newline character. If there are several possibilities, output the earliest one. The numbering of Professor Snørbørden's slides starts with zero (slide zero being the title slide and even this one being occasionally flawed).

3.4 Sample Data

Input	Output
3 5 1 1 0 0 0	2
5 7 1 0 0 1 0 0 1	1
5 15 0 1 1 1 0 0 1 0 0 0 1 1 0 1 0	4

4 Bicoloring

4.1 Problem

In 1976 the "Four Color Map Theorem" was proven with the assistance of a computer. This theorem states that every map can be colored using only four colors in such a way that no region is colored using the same color as a neighbor region. Here you are asked to solve a simpler similar problem. You have to decide whether a given arbitrary connected graph can be bicolored. That is, if one can assign colors (from a palette of two) to the nodes in such a way that no two adjacent nodes have the same color. To simplify the problem you can assume:

- There are no loops, i.e. no node has an edge to itself.
- There are no multi-edges, i.e. from any node to any other node there is at most one edge.
- The graph is undirected.
- The graph is strongly connected. That is, for each pair a, b of nodes, there exists a path from a to b in the graph.

The nodes of the graphs are identified by the integers 0 to $n-1$ for some n.

4.2 Input

The first line contains an integer n, specifying the number n of nodes in the graph ($1 < n \leq 200$). The second line contains the number m of edges ($1 \leq m \leq \frac{n(n-1)}{2}$). m lines follow, each containing two integers a and b ($1 \leq a, b \leq n$), specifying that there is an (undirected) edge between the nodes a and b.

4.3 Output

Output a single line containing the literal string *"BICOLORABLE."* or *"NOT BICOLORABLE."* (without the quotes), specifying whether the graph is bicolorable or not. Terminate the output with a single newline character.

4.4 Sample Data

Input	Output
3 3 0 1 1 2 2 0	NOT BICOLORABLE.
9 8 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8	BICOLORABLE.

5 Lawnmower

5.1 Problem

Source: Google Code Jam 2013, Qualification Round, licensed under Creative Commons Attribution[1].

Alice and Bob have a lawn in front of their house, shaped like an n by m meter rectangle. Each year, they try to cut the lawn in some interesting pattern. They used to do their cutting with shears, which was very time-consuming; but now they have a new automatic lawnmower with multiple settings, and they want to try it out.

The new lawnmower has a height setting - you can set it to any height h between 1 and 100 millimeters, and it will cut all the grass higher than h it encounters to height h. You run it by entering the lawn at any part of the edge of the lawn; then the lawnmower goes in a straight line, perpendicular to the edge of the lawn it entered, cutting grass in a swath 1m wide, until it exits the lawn on the other side. The lawnmower's height can be set only when it is not on the lawn.

Alice and Bob have a number of various patterns of grass that they could have on their lawn. For each of those, they want to know whether it's possible to cut the grass into this pattern with their new lawnmower. Each pattern is described by specifying the height of the grass on each 1m $\times$ 1m square of the lawn.

The grass is initially 100mm high on the whole lawn.

5.2 Input

The first line of the input contains two integers n and m, specifying the size of the lawn in meters ($1 \leq n, m \leq 100$). Next follow n lines, with the i-th line containing m integers $a_{i,j}$ each, the number $a_{i,j}$ describing the desired height of the grass in the j-th square of the i-th row.

5.3 Output

Output one line containing either the word *"YES"* (without the quotes) if it is possible to get the given pattern using the lawnmower, or *"NO"* (without quotes), if this is impossible.

[1] http://creativecommons.org/licenses/by/3.0/

5.4 Sample Data

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>3 3
2 1 2
1 1 1
2 1 2</td><td>YES</td></tr>
<tr><td>5 5
2 2 2 2 2
2 1 1 1 2
2 1 2 1 2
2 1 1 1 2
2 2 2 2 2</td><td>NO</td></tr>
<tr><td>1 3
1 2 1</td><td>YES</td></tr>
</table>

6 Elevator

6.1 Problem

Edward works as an engineer for the company *Lloyd Elevators & Lifts Incorporated* (*LEL Inc.*). His most recent assignment is to design a brand new elevator for a skyscraper.

Edward has a novel idea for how to navigate the elevator: He thinks that four buttons are enough to control the movement of the elevator. His proposal suggest the following four buttons:

- Move a floors up.
- Move b floors up.
- Move c floors up.
- Return to the first floor.

Initially, the elevator is on the first floor. A passenger uses the first three buttons to reach the floor they need.

Very soon after the announcement, LEL Inc. receives a lot of inquiries by people that want to know whether they will still be able to reach their apartments from the first floor after the change. They hired you to help them answer the requests.

6.2 Input

The first line of the input file contains three integers a, b, c — the parameters of the buttons ($1 \leq a, b, c \leq 10^5$). The second line contains an integer q — the number of inquiries ($1 \leq q \leq 10^5$). The next q lines contain an integer each. The i-th of these lines contains x_i, the floor that inquiry i is about ($1 \leq i \leq q, 1 \leq x_i \leq 10^{12}$).

6.3 Output

Output q lines, the answers to all the inquiries. Line number i of the output should contain the literal string *"YES"* (without quotes) if floor x_i can be reached from floor 1 or *"NO"* (without quotes) if this is not the case ($1 \leq i \leq q$). Terminate every line of output (including the last one) with a newline character.

6.4 Sample Data

Input	Output
3 4 8 7 1 2 3 4 5 6 7	YES NO NO YES YES NO YES
1 1 1 1 1000000	YES