

0 SimpleSum

0.1 Problem

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

0.2 Input

The first line contains a single integer n ($1 \leq n \leq 1000$). n lines follow, with the i th line containing a single integer representing a_i ($0 \leq a_i \leq 100$).

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, Unsolvble). 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 $\{1, \dots, 1000000\}$ (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 $\{i, i + 1, \dots, j - 1, j\}$, 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	29
6 5	99
1 100	27265634
6423 4245	

3 Speeding

3.1 Problem

Always the troublemaker, Bessie the cow has stolen Farmer John's tractor and taken off down the road!

The road is exactly 100 miles long, and Bessie drives the entire length of the road before ultimately being pulled over by a police officer, who gives Bessie a ticket for exceeding the speed limit, for having an expired license, and for operating a motor vehicle while being a cow. While Bessie concedes that the last two tickets are probably valid, she questions whether the police officer was correct in issuing the speeding ticket, and she wants to determine for herself if she has indeed driven faster than the speed limit for part of her journey.

The road is divided into N segments, each described by a positive integer length in miles, as well as an integer speed limit in the range $1 \dots 100$ miles per hour. As the road is 100 miles long, the lengths of all N segments add up to 100. For example, the road might start with a segment of length 45 miles, with speed limit 70, and then it might end with a segment of length 55 miles, with speed limit 60.

Bessie's journey can also be described by a series of segments, M of them. During each segment, she travels for a certain positive integer number of miles, at a certain integer speed. For example, she might begin by traveling 50 miles at a speed of 65, then another 50 miles at a speed of 55. The lengths of all M segments add to 100 total miles. Farmer John's tractor can drive 100 miles per hour at its fastest.

Given the information above, please determine the maximum amount over the speed limit that Bessie travels during any part of her journey.

3.2 Input

The first line of the input contains N and M , separated by a space.

The next N lines each contain two integers describing a road segment, giving its length and speed limit.

The next M lines each contain two integers describing a segment in Bessie's journey, giving the length and also the speed at which Bessie was driving.

3.3 Output

Please output a single line containing the maximum amount over the speed limit Bessie drove during any part of her journey. If she never exceeds the speed limit, please output 0.

3.4 Sample Data

Input	Output
3 3 40 75 50 35 10 45 40 76 20 30 40 40	5

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 ($0 \leq a, b < n$), 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 bicolored 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 Sum2D

5.1 Problem

You are given a rectangular field of integers with a width W and a height H . These integers are all in the range $[-1000,1000]$. Your task is to write a program that computes the sum of all integers in a rectangular subfield. You are therefore additionally given a list of Q subrectangles. For each of these subrectangles you should compute the sum.

5.2 Input

The first line contains W , H and Q in that order.

The next H lines contain each W integers separated by a space. This is the rectangular field.

After that another Q lines follow with 4 integers x_1, y_1, x_2, y_2 in that order. (x_1, y_1) and (x_2, y_2) form two opposing edges of a subrectangle. Coordinates are in the range $[1, W]$ or $[1, H]$. The edges should be included in the subrectangle.

H and W will not exceed 1000.

5.3 Output

For each subrectangle print the sum of the integers in it on a line of its own.

5.4 Sample Data

Input	Output
5 4 3	42
1 5 -3 8 -4	1
2 4 -1 -5 9	-9
10 1 0 -8 5	
3 5 9 0 1	
1 1 5 4	
2 3 2 3	
4 3 2 2	

6 Fish

6.1 Problem

In a small coastal country, all towns are situated on a long coastline (which we will model as a straight line). A long straight road runs along the coast, connecting the towns. The position of each town can be described by a single non-negative integer, the distance (in kilometers) from the start of the road.

Most of the citizens are fishermen, and they catch huge amounts of fish. After the fishing season is over and before the tourist season starts, the fish can be transported between different towns. A town can accommodate x tourists if it has x tons of fish available. The goal is to accommodate the largest possible number of tourists while distributing them evenly between towns. In other words, we want to find the largest integer y for which it is possible to distribute fish so that each town can accommodate at least y tourists.

In one shipment, an integral number of tons of fish is sent from one town to another. During transportation, one ton of fish per kilometer traveled is lost to hungry pillagers descending from the mountains. More formally, if a town ships f tons of fish to another town that is d kilometers away, then $f - d$ tons will arrive at the destination; if f is less than d , then the entire shipment is lost.

It is possible to arbitrarily repackage and combine shipments in intermediate towns. For example, we can send shipments from towns a and b to town c , combine half of the remaining fish from both shipments with the fish originating in c and send it in a single large shipment from town c to town d .

Write a program that, given the positions of all towns and the amount of fish each town produces, determines the largest number of tourists that can be accommodated by each city after the fish have been distributed.

6.2 Input

The first line contains a single integer n , $1 \leq n \leq 10^5$, the number of towns. Each of the following n lines contains two integers p and f , $0 \leq p, f \leq 10^{12}$, the position of a town (in kilometers) and the amount of fish it produces (in tons). The towns will be sorted in ascending order of position. The positions of all towns will be distinct.

6.3 Output

Output the largest number of tourists y from the task description followed by a newline character.

6.4 Sample Data

Input	Output
5 1 0 2 21 4 0 5 7 7 6	6

7 Elevator

7.1 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.

7.2 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 ($1 \leq i \leq q, 1 \leq x_i \leq 10^{12}$).

7.3 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.

7.4 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