0 SimpleSum

0.1 Problem

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

0.2 Input

The first line contains a single integer n. The next n lines each contain a single integer representing a_i.

0.3 Output

Output the sum s followed by a newline character.

0.4 Sample Data

Input

```
3
1
2
42
```

Output

```
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 any two numbers i and j you are to determine the maximum Collatz-length of numbers in $[i, j]$.

1.2 Input

The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 1000 and greater than 0. You should process all pairs of integers and for each pair determine the maximum Collatz-length over all integers in $[i, j]$. No operation overflows a 32-bit integer.

1.3 Output

For each pair of input integers i and j you should output i and j followed by the corresponding maximum Collatz-length. These three numbers should be written into a single line, in that order, separated by a single space. Each line must be terminated by a newline character.

1.4 Sample Data

Input

```
1 10
100 200
201 210
900 999
```

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 input consists of several testcases (at most 100000). Each testcase is on a line of its own and consists of the two integers $1 \leq w \leq 2^{13}$ and $1 \leq h \leq 2^{13}$.

2.3 Output

For every case in the input, output one line containing the smallest number of breaks. Each line should be terminated by a newline character.

2.4 Sample Data

Input

```
6 5
1 100
6423 4245
```

Output

```
29
99
27265634
```

3 Maze

3.1 Problem

While eagerly studying for the final exams, an enthusiastic computer science student was haunted by the weird dream of being trapped in a screensaver one night. He was surrounded by brick walls forming a complicated maze and started looking for an exit. Just as he began wondering if there was any exit at all, he awakened soaked in sweat and immediately started to try to solve the problem using his beloved computer.

3.2 Input

The input starts with three lines containing the maze's width w ($3 \leq w \leq 100$) and height h ($3 \leq h \leq 100$), the start position x_s, y_s and the exit position x_e, y_e. The points in the maze are indexed by a coordinate system which sits at the maze's top left corner with the x-direction pointing right and the y-direction downward.

The remaining h lines contain a depiction of the maze where a 1 as x-th number of line y means that the position x, y is blocked by a wall (we count from 0). Free tiles are denoted by a 0.

The maze is guaranteed to be surrounded by a wall and neither the start nor the exit position will be a wall tile.

3.3 Output

You should print "Y" if the exit position is reachable from the start position by only moving horizontally and vertically; else "N".

3.4 Sample Data

Input

```
5 4
1 1
3 1
1 1 1 1 1
1 0 1 0 1
1 0 0 0 1
1 1 1 1 1
```

Output

```
Y
```

4 Light Bulbs

4.1 Problem

To make the lives of their students simpler, the Unseen University decided to demolish their entire infrastructure and replace it with a single central building. This building consists of infinitely many rooms labelled 1, 2, et cetera, each of which contains exactly one light bulb.

To save costs, the planing committee chose to spend the amount of money which had to be used for art on a fancy lighting system. The artists responsible for the project constructed a nice pattern for turning the light bulbs in the rooms on and off: At the beginning all the lights are switched off, then the lights in every room are toggled, then in every second room (rooms 2, 4, 6, ...), in every third (rooms 3, 6, 9, ...) and so on. Toggling a light bulb means switching it from on to off and vice versa.

Although the employees of the university are delighted about the enrichment of their workday by art, they are also concerned that they are going to sit in the dark after the project has run its course. So they decide that everyone should move to a room which will be lit after the whole pattern has been completed. To not cause undue confusion, they want to remain in the same order, i.e. the occupant of room i should move to the i-th lit room. Note that although the whole lighting pattern has an infinite number of steps, a given light bulb is only affected by a finite number of them. So we can unambiguously say whether it will be switched on or off at the end.

Can you help them determine which rooms they should move to?

4.2 Input

The input contains the number of test cases nr ($1 \leq nr \leq 100000$) on the first line. For each test case a line containing an original room number n ($1 \leq n \leq 2^{13}$) follows.

4.3 Output

For each test case, print the number of the n-th lit room on a separate line.

4.4 Sample Data

Input

```
3
1
67
2485
```

```
Output
1
4489
6175225
```

5 Domino

5.1 Problem

You are given a chessboard of size $2 \times n$ and n indistinguishable 2×1 dominoes. Determine the number of possibilities to fill the board with dominoes without overlaps. As the number of possibilities can become very large, you should output the result modulo $p = 999999001$.

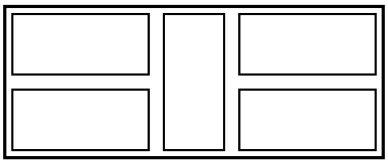

Figure 1: Valid filling of a 2×5 grid.

5.2 Input

The input contains the number of test cases nr ($1 \leq nr \leq 50000$) on the first line. For each test case a line containing exactly one integer, the number n ($1 \leq n \leq 10^6$), follows.

5.3 Output

For each test case you should output the number of possibilities modulo p on a line of its own.

5.4 Sample Data

Input

```
4
42
9
333333
1000000
```

Output

```
433494437
55
399696940
745641431
```

6 Haters gonna hate

6.1 Problem

The Dean of the Faculty of Computer Science, the generous man he is, invites all of his students to a banquet in his mansion at the end of every winter term. All the students show up, not so much for the anecdotes the Dean shares with the attendants, but for the free beer and, above all, to see the beautiful daughters of the Dean.

Since the number of female computer science students is negligible, as is the number of male ones with girlfriends (we're talking about a nerdy subject after all), it is fair to say that every student attending the banquet has made up his mind about the daughters.

There are two types of students:

Lovers: A lover is a student that has a crush on exactly one of the daughters.

Haters: A hater is a student that, the bitter and spoiled person he is, rages about how much of a stupid *lady*[1] a certain daughter is.

Standing in a queue in front of the Dean's house waiting for the Dean to let them in, every student is chatting with the student directly in front and directly behind him. Though computer science students tend to be peaceful in general, in case a hater and a lover with respect to the same daughter stand next to each other, they see no choice but to settle their dispute like real men: Go to their respective homes, meet on a neutral server and frag each other to bits, until both of them realise that gaming is better than girls anyway. Whenever two students wander off this way, the adjacent students close the gap and continue to chat.

The Dean wants to set the table and therefore needs to know how many students will be attending his banquet, that is, how many people will be waiting in front of the house when he opens the door.

6.2 Input

The first line of the input contains an integer denoting the number of test cases c that follow ($0 \leq c \leq 100000$). Each test case consists of one line containing the number of students n in the queue ($0 \leq n \leq 100000$) and n integers $s_i \in \{-127, -126, \ldots 127\} \setminus \{0\}$ describing each student ($1 \leq i \leq n$).

A positive number means the student is a lover, a negative number means the student is a hater. The absolute value denotes the daughter (for some ordering of the daughters).

[1] redacted

6.3 Output

For each test case, output the number of students that will be left in the queue once all conflicts of neighbouring students have been resolved on a separate line.

6.4 Sample Data

Input

```
5
6 1 2 3 -3 4 5
10 -1 -2 -3 -4 -5 5 4 3 2 1
3 1 1 1
0
9 5 4 -4 4 6 7 -6 -7 1
```

Output

```
4
0
3
0
7
```

7 Sum2D

7.1 Problem

You are given a rectangular field of width W and height H that contains integers. These integers are all in the range $[-1000, 1000]$. Your task is to write a program that computes the sum of all integers in Q given subrectangles of the field.

7.2 Input

The first line contains $1 \leq W \leq 1000$, $1 \leq H \leq 1000$ and $1 \leq Q \leq 150000$, in that order.

H lines each with W integers, separated by a space, follow. This is the field.

After that another Q lines follow, each contains 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 $[1, W]$ and $[1, H]$, respectively. The edges should be included in the subrectangle.

7.3 Output

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

7.4 Sample Data

Input

```
5 4 3
1 5 -3 8 -4
2 4 -1 -5 9
10 1 0 -8 5
3 5 9 0 1
1 1 5 4
2 3 2 3
4 3 2 2
```

Output

```
42
1
-9
```

8 Fish

8.1 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.

8.2 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.

8.3 Output

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

8.4 Sample Data

Input

```
5
1 0
2 21
4 0
5 7
7 6
```

Output

```
6
```