ECOO 2018

Programming Contest Questions

Local Competition (Round 1)

March 21 – March 28, 2018

Problem 1: Willow's Wild Ride

Mandy is a working artist and is commissioned to create some art pieces for 2018. The client wants Mandy to create art pieces out of cardboard boxes to display at the local art show. Occasionally, Mandy plans to drive by the local store on her way home to grab some boxes for her art pieces.

Mandy's cat, Willow, likes to play with every box that Mandy brings home. Willow plays with a box for **T** days before getting bored of it. Once Willow is bored with a box, she never returns to it again, meaning that Mandy can finally use the empty box in her art projects.

If Mandy brings home another box before Willow finishes playing with the previous one, Willow will wait until she is bored with the previous box before moving onto the new one.

Given Mandy's box-shopping habits over the next **N** days, can you determine by how many days the project will be delayed due to Willow?

Input Specifications

DATA11.txt (DATA12.txt for the second try) will contain 10 datasets.

Each dataset begins with two integers T ($2 \le \mathbf{T} \le 7$) and N ($1 \le \mathbf{N} \le 365$). The next **N** lines each contain either the letter 'E' or 'B' which represent whether Mandy came home empty-handed or with a box that day.

Output Specifications

For each dataset, output the number of days that Willow will be playing with the boxes after the **N** days given in the dataset.

Sample Input (Two Datasets Shown)

```
3 5
E
B
E
B
E
2 4
B
E
E
E
```

Sample Output

```
2
0
```

Problem 2: Rue's Rings

A local street planning consulting company, Rue's Rings, is looking to help the city prepare for converting a lot of their medium-traffic one-lane intersections into roundabouts.

After generating a plan, the city wants to run a simulation to see where the possible bottlenecks of traffic could be. The simulation runs by finding the roundabouts along a route and then figuring out which roundabout is the smallest in diameter. The smallest diameter roundabout would be the best possible location for congestion and creating a bottleneck of traffic which would make the traffic patterns even worse than they are now.

There are **N** roundabout-filled routes from the starting point to the endpoint. Your task as the city simulation specialist, is to analyze the different routes that are available to find out which route (or routes) could generate the most issues.

Input Specifications

DATA21.txt (DATA22.txt for the second try) will contain 10 datasets. Each dataset begins with an integer **N** (2 ≤ **N** ≤ 700), the number of routes. The next **N** lines each contain a series of integers describing a route.

The first integer of each route description is the ID for the route. The second integer **R** (1 ≤ **R** ≤ 70) is the number of roundabouts along the route. **R** integers follow which describe the diameter **D** (1 ≤ **D** ≤ 70,000) of each roundabout along the route.

Output Specifications

For each dataset, output the minimum roundabout diameter along a route followed by a brace-enclosed, sorted list of route IDs for the routes that could cause issues.

Sample Input (Two Datasets Shown)

```
3
1 6 4 5 2 6 3 2
2 3 2 3 4
3 4 2 3 2 4
4
1 2 3 4
2 3 4 2 4
3 7 2 3 3 4 5 2 6
4 5 3 2 5 1 4
```

Sample Output

```
2 {1,2,3}
1 {4}
```

Problem 3: Missing Art

The local museum had a possible theft in their displays recently. The thief would have replaced the original art piece with a similarly created replica. The museum has a security system in place that is invisible to the naked eye, and they want to ensure that the art that is in the case is the original art, and not the replica.

For security, the **N** display cases have a specific digital code overlaid on the glass case that is invisible to the naked eye. The digital code is only readable by a scanner to tell the curators whether or not an art piece is an original. If someone were to tamper with the glass in any way, there would be an issue with the digital code.

The codes that are on the display must be converted to the values that the scanner has in its database. Converting the digital codes to the scanner-readable codes is done as follows:

- Each even digit must be increased by **X**. Digits can increase above 9, in which case they are replaced with the two-digit result.
- Each odd digit must be decreased by **Y**. Digits cannot decrease below 0.
- Each digit that equals zero must be changed to **Z**

You will also be given the expected scanner-readable codes for each of the art display cases, so that you can check to see if they match.

Your task is to find out whether the piece of art is authentic, and if it isn't, then you must "show your work" so that the curators can see the proof of your work.

Input Specifications

DATA31.txt (DATA32.txt for the second try) will contain 10 datasets. Each dataset begins with four integers **N**, **X**, **Y**, **Z** (5 ≤ **N** ≤ 20, 1 ≤ **X**, **Y**, **Z** ≤ 10). The next **N** lines each contain a display's digital code (of length at most 100 digits). The next line contains the letter 'A' signifying the start of the answer key. The **N** lines each contain the answer keys to the display cases' digital codes. Finally, each dataset ends with a line containing an asterisk (*).

Output Specifications

For each dataset, if all the digital codes match with the answer keys, output "MATCH". Otherwise, output "FAIL: " followed by a comma-separated list of which display cases had a mismatch. Display cases are numbered starting from 1.

(Sample datasets shown on next page)

Sample Input (Two Datasets Shown):

```
4 1 2 3
12345678
87654321
28374827
18723470
A
72384728
72374822
39155935
32874281
*
3 2 3 1
02934820
40294838
82934820
A
146061041
6146610010
1046061041
*
```

Sample Output:

```
FAIL: 1,2,4
MATCH
```

Problem 4: Fibonacci Spiral

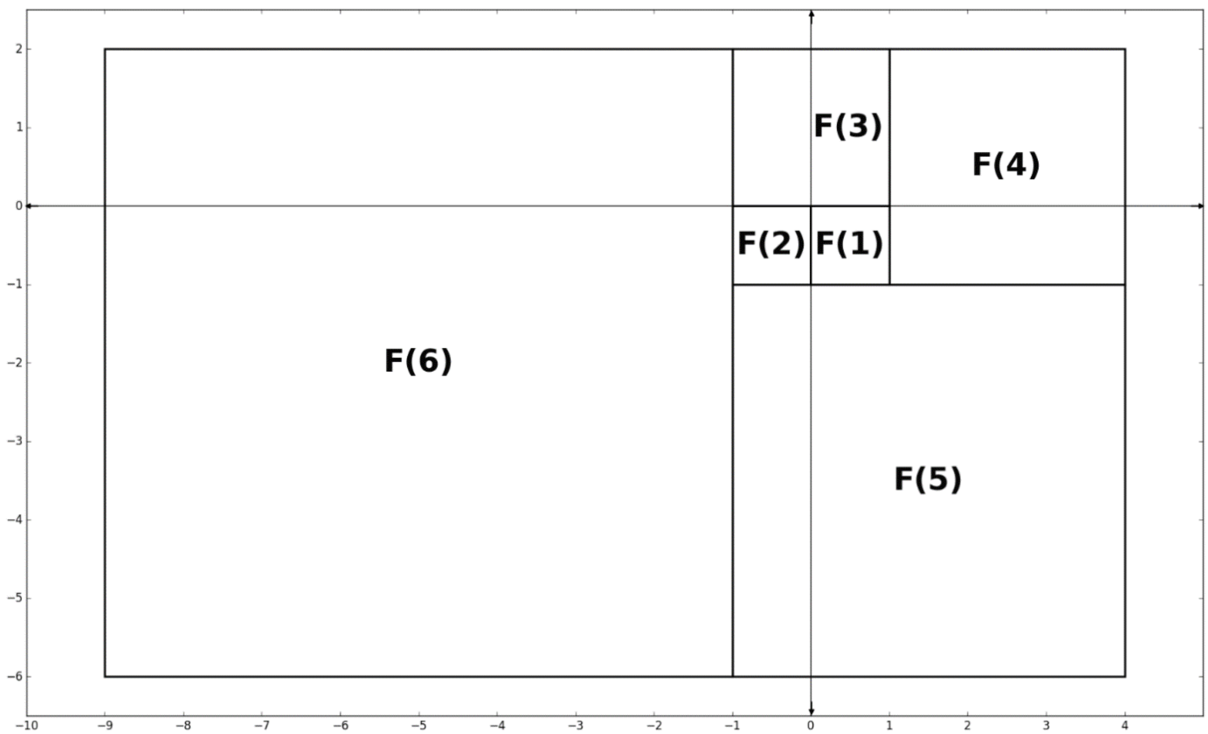
The Fibonacci sequence can be represented as follows:

- The first two Fibonacci numbers, F(1) and F(2), are both equal to 1.
- For N > 2, the Nth Fibonacci number is the sum of the two previous ones, i.e.

$$F(N) = F(N-1) + F(N-2)$$

The first few terms of the sequence are 1, 1, 2, 3, 5, 8, 13, ...

An interesting way to represent this sequence is by starting as a square with a side length equal to F(1) and then repeatedly adding squares in clockwise order with the side length of the Nth square being equal to the F(N). By doing this, a rectangle will always be formed. Here is a diagram:



This process can be repeated indefinitely, which means that we can cover the plane using these Fibonacci squares. Suppose we repeat the process as shown in the diagram above with the top-left corner of the first square being the origin (0, 0).

Given a coordinate (**X**, **Y**), can you figure out which Fibonacci square the coordinate is in?

Input Specifications

DATA41.txt (DATA42.txt for the second try) will contain 10 datasets. Each dataset consists of two space-separated integers X, Y ($-10^9 \le$ **X**, **Y** $\le 10^9$) representing a point on the plane.

(Continued on next page)

Output Specifications

For each dataset, output the smallest integer N such that the point (**X**, **Y**) is contained (or is on the boundary of) a square of side length F(N).

Sample Input (5 Datasets Shown)

```
0 0
1 1
2 1
3 -2
-9 0
```

Sample Output

```
1
3
4
5
6
```


ECOO 2018 Question Development Team

Kevin Forest ....Sheridan College
John Ketelaars .... ECOO-CS Communications
Stella Lau ....University of Cambridge
Greg Reid ....St. Francis Xavier Secondary School, Mississauga
Sam Scott ....Mohawk College
Andrew Seidel .... John Fraser Secondary School, Mississauga
David Stermole .... ECOO-CS President
Reyno Tilikaynen .... University of Waterloo
Andy Huang .... University of Waterloo