TJ IOI 2018
Programming Round

Thomas Jefferson High School for Science and Technology

Saturday, April 28, 2018

Instructions

1. The following section consists of 10 problems, arranged in approximate order of difficulty. You will have 3 hours to complete as many as you can.
2. Each problem consists consists of 10 sample input cases of increasing complexity. 1 point is awarded for each input case correctly solved, for a total of 10 points per problem, and 100 points for the entire round.
3. The first test case will always be the sample case. We may disqualify any programs that consist of only print statements.
4. Programs must read from standard input and print to standard output. Ensure that your program does not print any extraneous output, such as debug statements.
5. Unless otherwise stated, programs must run in 2 seconds and in 256 MB of memory.
6. Accepted languages include Java 8, Python 3, and C/C++. You must submit the source code of your programs, not executable files.
7. For Java programs, the name of the Java class and the file name should be the short name. For example, a solution for the problem with the short name `test` must be in a class named `test`, in a file named `test.java`.
8. In an effort to give back feedback as soon as possible, automated responses are given for each submission. Feedback will consist of success or failure, along with reason for failure if applicable. We reserve the right to make final decisions on judging.
9. If you have a question concerning a problem, you may submit a request for clarification through the grader interface.
10. You are permitted to use any printed material that you have brought with you, including the official TJ IOI Study Guide. You are also permitted to use the electronic documentation provided to you through the contest site. You many not collaborate with anyone outside of your team.
11. Each team may use only one computer and all code must be written and submitted through this computer. If you are experiencing technical issues, alert a proctor and we will assist you.
12. You are not permitted to access the Internet in any way other than accessing the contest site. You may not use any electronic or communications devices other than the computer. You may not intentionally exit the virtual machine for any reason. You may not attempt to attack or damage the grader. Breaking any of these rules may be grounds for disqualification.
13. Good luck and have fun!

Do not turn the page until instructed to do so.

Contents

A Driving

Daniel recently got his driver's license!
Unfortunately for him, he doesn't have a GPS (or a smartphone, for that matter). Instead, he relies on the order of the road signs he passes to find his way to his destination. Each road sign only has a single lowercase letter on it.
Wassim, in his ever-lasting quest to deceive Daniel, decides to change a few road signs to confuse him. He hopes to make the road signs such that they come in the same order forwards and backwards, so that Daniel doesn't know which way to go.

Given the initial ordering of the road signs, denoted by a string of lowercase letters s, determine the minimum signs that Wassim must change to make them read the same forwards and backward.

SHORT NAME: `driving`

INPUT FORMAT:
The first line of input contains the string s. The length of s will be between 1 and 10^5, inclusive.

OUTPUT FORMAT:
Output an integer k that denotes the minimum number of characters that must be changed in s to make it a palindrome.

SAMPLE INPUT:

```
awejsewz
```

SAMPLE OUTPUT:

```
2
```

We can change the string to "zwessewz" with two edits, making it a palindrome.

B Attendance Fraud

At TJ, everyone wants to be first in everything, including attendance. Some students have started changing their names just to be first! To make sure the teachers don't notice, students only perform a cyclic shift of their name. For example, justin could make his name justin, ustinj, stinju, tinjus, injust, or njusti. Given a student's name, print out the version that appears first in alphabetical order.

SHORT NAME: `fraud`

INPUT FORMAT:
The input will be one string, consisting of lowercase characters a-z, followed by a newline. The string will contain less than 2000 characters.

OUTPUT FORMAT:

Print the version of the given name that the student should use.

SAMPLE INPUT:

```
hannah
```

SAMPLE OUTPUT:

```
ahhann
```

C Treasure

Jimmy recently found a treasure "map" (an integer N). After much thinking, he's realized how to decode N to find the location of the treasure.
The scheme is as follows: starting at (0,0), he must divide N by 4 (integer division) and act based on the remainder. If it's 0 - go North, 1 - go East, 2 - go South, 3 - go West. Jimmy must then repeat this with the quotient until he gets 0, after which he has found the correct location.

Please help Jimmy decode his map and find the treasure!

SHORT NAME: `treasure`

INPUT FORMAT:
The first line of input contains the integer N, the map number. ($1 \leq N \leq 10^5$)

OUTPUT FORMAT:
Output two integers separated by a comma and a space which represent the location of the treasure.

SAMPLE INPUT:

```
5
```

SAMPLE OUTPUT:

```
2, 0
```

$5 \div 4$ has a remainder of 1, so he must go east. The quotient, after integer division, is also 1.
$1 \div 4$ has a remainder of 1, so he must again go east. The quotient is 0, so he's done.

D Stock Market

Jaden recently starting trading in the stock market!
His strategy is as follows: on day 0, invest in N stocks ($1 \leq N \leq 100$), allocating a certain amount of money to each of them. Then, like any diligent trader, he notes the daily percentage change of each of these N stocks for the next K days ($1 \leq K \leq 50$).
Over these K days, does not make any additional trades (in fact, he uninstalls his broker app).
Now that these K days have passed, Jerry wants to figure out Jaden's net profit. Unfortunately for him, Jaden is rather secretive of his portfolio. Instead of giving Jerry his initial allocation of money for the N stocks that he decided upon on day 0, he gives Jerry C possible initial allocations ($1 \leq C \leq 5000$), one of which is his actual allocation (the other allocations are from his friends, who tried to copy his strategy).
Jerry is quite certain that the allocation that had the highest net profit is Jaden's actual allocation. Under this assumption, find the amount of money Jaden gained (or lost, as the case may be).
Please print this value *rounded to the nearest integer*.

This problem is batched. In 70% of the test cases, $C \leq 50$.

SHORT NAME: `stocks`

INPUT FORMAT:
The first line of input contains the integers N and K, the number of stocks and the number of days.
The second through $1 + K$th lines each contain N decimals, the ith of which denotes the percentage change of the ith stock for the day. That is, the second line denotes the daily percentage change for each stock on day one, the third line denotes the daily percentage change for each stock on day two, and so on.
The third line contains an integer C, denoting the number of allocations.
The fourth through $3 + C$th lines each contain N decimals (between 1 and 1000), denoting a single possible allocation.

Note: the daily percentage change for a given stock will never exceed 20% in absolute value.

OUTPUT FORMAT:
Output a single integer denoting the amount Jaden has gained (negative if he lost money), rounded to the nearest integer, under the assumption that the most profitable allocation was his actual allocation.

SAMPLE INPUT:

```
2 3
-18.7 2.2
-10.1 -6.3
5.8 0.0
3
950.2 305.72
5.0 5.0
```

```
306.1 406.16
```

SAMPLE OUTPUT:

```
-1
```

The optimal configuration is the second, in which he invests $5 in both stocks A and B.
For stock A, he ends up with $3.87.
For stock B, he ends up with $4.79.
This totals to $8.65, a loss of $1.35.

E Bridges

Oh no! The recent earthquake that hit Treeville destroyed many bridges that link the trees of the city. Your job is to find the minimum number of bridges that need to be rebuilt such that no part of Treeville is isolated from any other part.

Treeville has N ($1 \leq N \leq 10^5$) trees, numbered 1 through N. A bridge connects one tree to another. Because the earthquake was so devastating, there is no more than M ($0 \leq N \leq 10^5$) bridges after the earthquake.

SHORT NAME: `bridges`

INPUT FORMAT:
The first line of input contains N, the number of trees in Treeville and M, the number of bridges after the earthquake.

The second through M+1 lines of input each contain two integers, a and b, meaning there is a bridge between tree a and tree b.

OUTPUT FORMAT:
A single integer, specifying the minimum number of bridges that need to be built so no part of Treeville is disconnected from another part.

SAMPLE INPUT:

```
5 3
1 2
2 3
4 5
```

SAMPLE OUTPUT:

```
1
```

F Census

In the nation of Vserossijskaja, a census was recently taken. In Vserossijskaja, the N ($1 \leq N \leq 2*10^5$) cities are named after numbers $a_1, a_2, \ldots, a_n$ (for any a_k, $1 \leq a_k \leq 2*10^5$). Coincidentally, if a city is named a_k, the city has a_k streets, numbered from 1 to a_k. The jth street in the city named a_k has $\lfloor \frac{a_k}{j} \rfloor$ houses, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

For example, a city named 5 would have $\lfloor \frac{5}{1} \rfloor + \lfloor \frac{5}{2} \rfloor + \lfloor \frac{5}{3} \rfloor + \lfloor \frac{5}{4} \rfloor + \lfloor \frac{5}{5} \rfloor = 5+2+1+1+1 = 10$ houses.

Vserossijskaja has decided to implement a new milk delivery service, which is most efficient when there are closest to K houses in a city.

Given N and K, determine the city which will be best suited for the new milk service; that is, the city with a number of roads closest to K.

SHORT NAME: `census`

INPUT FORMAT:
Line 1 contains the number of cities N followed by the optimal number of houses K
Line 2 through $N+1$ contain a_k, the names of the different cities

OUTPUT FORMAT:
Output an integer representing the name of the optimal city.

SAMPLE INPUT:

```
4 10
4
9
8
2
```

SAMPLE OUTPUT:

```
4
```

City 4 has $\lfloor \frac{4}{1} \rfloor + \lfloor \frac{4}{2} \rfloor + \lfloor \frac{4}{3} \rfloor + \lfloor \frac{4}{4} \rfloor = 4+2+1+1 = 8$ roads. Of the four cities, this is closest to 10.

G Merry-Go-Round

While hundreds of joyful children ride a merry-go-round, you have to analyze it. The merry-go-round has N horses, each with some number of children riding it. As the merry-go-round rotates, all the kids get off the first horse. One kid from the first horse will get onto the second, one onto third, and so on until there are no kids left on the first horse. Generally, if K children get off horse a, 1 child will get onto each of the K horses after a. If there are more than N kids on the horse, then one kid will get onto each horse, including the one they just dismounted, and the extras will go home. The merry-go-round will then rotate by one horse so that all the kids on the second horse will get onto other horses as described above. This process will repeat forever.

Eventually, the same configuration of children will repeat. A configuration is considered the same if there are the same number of kids on each horse. All the horses are indistinguishable, so we compare configurations starting from the next horse in each, instead of tracking the original first horse.

From a starting configuration, the configuration of children will eventually repeat. After the configuration repeats, it will stay in a cycle that infinitely repeats the same sequence of configurations. Given the initial configuration of children, find the period of the cycle. Also find the number of configurations, possibly including the initial state, that will never be repeated.

SHORT NAME: `merry`

INPUT FORMAT:

The first line of input contains N, the number of horse on the merry-go-round a ($1 \leq N \leq 150$).

The second line contains the number of children on each horse, starting from the next horse ($1 \leq a_i \leq 150$), separated by spaces.

OUTPUT FORMAT:

First print the number of steps in the cycle, followed by a space, and then the number of configurations that are never repeated.

Note: You can assume that both answers will be less than 20,000.

SAMPLE INPUT:

```
4
1 5 2 9
```

SAMPLE OUTPUT:

```
2 4
```

The sequence of configurations is:

```
1 5 2 9
0 6 2 9
1 1 3 10
2 2 0 11
3 3 1 1
0 4 2 2
1 1 3 3
```

Note that 3,3,1,1 and 1,1,3,3 are considered the same configuration, starting from the next horse.

H Cafeteria Chaos

As part of the recent renovation, TJ got a new cafeteria. This cafeteria offers N ($1 \leq N \leq 5000$) food options.
On the first day, there was complete chaos in the cafeteria. The designers had not realized that any student who eats the spicy curry will immediately have to find some milk. After drinking the milk, those students will need to find some cookies. It turns out that every food causes students to search for another food. Chips, for example, just make students get more chips. For every minute of lunch that passes, each student will move on the the next food they want.

TJ has a famously long lunch period, K minutes long ($0 \leq K \leq 10^9$). For each food that students can begin lunch eating, find what food they will want at the end of lunch. Note that if lunch is zero minutes long, students will still want their first food.
SHORT NAME: `chaos`

INPUT FORMAT:
The first line contains N and K separated by a space.
The second line contains N integers, separated by spaces, ($0 \leq a_i < N$). This means that food i makes students want food a_i.
Note: For four out of ten test cases K will be less than 2000.

OUTPUT FORMAT:

Print what food students will end wanting for each food. Print each answer, starting from food 0, on its own line.

SAMPLE INPUT:

```
5 2
1 0 1 4 2
```

SAMPLE OUTPUT:

```
0
1
0
2
1
```

I Prime Cows

Farmer John is lining up his N cows for inspection!
Each of these N cows can be associated with any digit 0 through 9. Let's denote the digit of the ith cow to be c_i.
Farmer John determines that the cows will pass inspection if the F-digit number $c_1c_2c_3 \ldots c_F$ is prime, for *every* F such that $1 \leq F \leq N$.
Help the cows find the number of distinct ways they can assign digits to pass the inspection.

SHORT NAME: `prime`

INPUT FORMAT:
Line 1 contains the integer N, the number of cows.

OUTPUT FORMAT:
Output the number of ways the cows can be numbered to pass inspection.

SAMPLE INPUT:

```
2
```

SAMPLE OUTPUT:

```
4
```

The assignments that work in the sample case are 2|3, 3|7, 5|3, and 7|3.

J Maximal Non-Coprime

Given an integer array a of length N ($1 \leq N \leq 10^4$, $1 \leq a_i \leq 10^3$), perform Q queries ($1 \leq Q \leq 10^4$).
Queries will can be in the following formats:

- `C` i v: change the number at index i (0-indexed) to the value v ($1 \leq v \leq 10^3$).
- `F` i j v: find the maximal number k with index between i and j (inclusive) such that $gcd(k, v) \neq 1$. In other words, k is the maximum in $a[i...j]$ such that k shares a factor (other than 1) with v ($1 \leq v \leq 10^3$). If there is no such number, output -1.

Note: There are 168 primes less than 10^3.

SHORT NAME: `max_noncoprime`

INPUT FORMAT:
The first line of input contains N, the length of the array a ($1 \leq N \leq 10^4$).
The second line contains the elements of array a ($1 \leq a_i \leq 10^3$), separated by spaces.
The third line contains the number of queries Q ($1 \leq Q \leq 10^4$).
The fourth through $3 + Q$ lines (inclusive) contain each query. Queries are either of the format `C` i v or `F` i j v ($0 \leq i, j \leq N - 1$, $1 \leq v \leq 10^3$).

OUTPUT FORMAT:
For each query that starts with `F`, output the maximal integer k in $a[i...j]$ such that $gcd(k, v) \neq 1$, or -1 if k does not exist.

SAMPLE INPUT:

```
4
2 3 49 25
6
C 0 10
F 1 2 5
F 0 3 5
C 3 5
F 0 3 5
F 1 3 7
```

SAMPLE OUTPUT:

```
-1
25
10
49
```