TJ IOI 2017
Programming Round

Thomas Jefferson High School for Science and Technology

Saturday, May 13, 2017

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. 10 points are awarded for each input case correctly solved, for a total of 100 points per problem, and 1000 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 2017 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 Larry's Race

TJ IOI Inc. has chosen Larry as their corporate representative at the local track and field competition! However, the competition has a very peculiar set of rules: if Larry would like to advertise TJ IOI Inc., he must compete in the race! To get Larry in shape, Devon has built a robot to chase Larry, traveling 100 meters in T seconds ($1 \leq T \leq 100,000$).

There are N inputs to this problem ($1 \leq N \leq 100,000$). Each consists of a distance A_i that Larry runs, where A_i is a multiple of 100 ($100 \leq A_i \leq 100,000$), and the time B_i it took for him to run that distance ($1 \leq B_i \leq 100,000$), determine whether Larry could outrun Devon's robot.

Note: if Devon's robot catches Larry exactly at the finish line, Larry did not outrun it.

SHORT NAME: `race`

INPUT FORMAT:
The first line consists of two integers, N and T. The next N lines each contain an integer A_i representing a distance in meters (A_i is a multiple of 100), and a time B_i representing the time it took Larry to run that distance in seconds.

OUTPUT FORMAT:
For each input, if Larry outran Devon's robot, output "SPEEDRACER" (without quotes). Otherwise, output "POTATO" (without quotes).

SAMPLE INPUT:

```
3 22
1600 352
800 150
3200 840
```

SAMPLE OUTPUT:

```
POTATO
SPEEDRACER
POTATO
```

B Lunchbox Hunt

Devon has prepared an extravagant lunch to celebrate the one year anniversary of TJ IOI Inc.! However, to make things more interesting, he has hidden the lunchbox containing his lunch somewhere inside the huge, single-level parking garage.

Alex has decided to go on a treasure hunt to find Devon's hidden lunchbox. Fortunately for Alex, Devon has left behind a set of instructions specifying the location of his lunchbox. The instructions consist of a starting location (x_0, y_0) and N $(1 \leq N \leq 1,000,000)$ queries. Each query consists of a direction specified by the characters 'N', 'S', 'E', and 'W', and a non-negative distance. Help Alex find the coordinates of the location of Devon's lunchbox.

Note: Alex's position (x, y) at any time is guaranteed to remain within $-1,000,000,000 \leq x, y \leq 1,000,000,000$.

SHORT NAME: `lunchbox`

INPUT FORMAT:
The first line will contain three integers N, x_0, and y_0. The following N lines will describe a query consisting of a character ('N', 'S', 'E', 'W') and a non-negative integer distance.

Note: North corresponds to up, south corresponds to down, east corresponds to right, and west corresponds to left.

OUTPUT FORMAT:
The output should consist of two integers separated by a space. The first integer is the final x coordinate and the second integer is the final y coordinate.

SAMPLE INPUT:

```
4 6 -2
N 3
S 5
W 2
W 1
```

SAMPLE OUTPUT:

```
3 -4
```

C Singing Low

After the work day is over at TJ IOI Inc., many of the employees attend office karaoke! One of these employees is Devon, who wants to see how low he can sing. Devon begins at note N ($1 \leq N \leq 1,000,000$), and would like to sing note 0. In one step, Devon may sing between 1 and K notes lower than his current note. (For example, if K is 3 and he is on note 5, he can sing either 3, 2, or 1 notes lower than his current note, taking him to notes 2, 3, or 4, respectively.)

However, Devon's note cannot decrease by any given amount more than once. (For example, if he went from note 5 to note 3 in the previous example by going down 2 notes, he would not be able to go to 1 as this be another decrease by 2 notes.) Please help Devon calculate the smallest value of K that will allow him to get to note 0.

SHORT NAME: `singing`

INPUT FORMAT:
The first line will contain the integer N, the note that Devon begins on.

OUTPUT FORMAT:
The output should consist of one integer, K, the lowest value which will allow Devon to reach note 0.

SAMPLE INPUT:

```
8
```

SAMPLE OUTPUT:

```
4
```

D Pencils

TJ IOI Inc. has reached a net worth of a million dollars! Kevin, the CEO, is very happy of this achievement, and asks Alex to write a report on the financial standing of the company, first thing tomorrow. However, Kevin decides that instead of typing the report, he will make Alex hand write it, in order to build character.

That night, Alex has gathered N ($1 \leq N \leq 100,000$) pencils in front of him. However, his pencil bag only has room for K pencils ($1 \leq K \leq N$). Each of his pencils has an integer length in the range of 1 to $1,000$, inclusive. Because Kevin expects an extravagant and exhaustively detailed report, Alex will have to write a lot if he wants to please Kevin, so Alex wants to choose the longest K pencils to place into his pencil bag. Please help Alex determine the sum of the lengths of those pencils.

SHORT NAME: `pencils`

INPUT FORMAT:
The first line of input contains two integers N and K. The next N lines describe the length of the pencils.

OUTPUT FORMAT:
Output a single integer, the sum of the K longest pencils.

SAMPLE INPUT:

```
7 3
1
4
5
3
8
14
2
```

SAMPLE OUTPUT:

```
27
```

E Puck Puck Moose

TJ IOI Inc. is having its annual corporate retreat to build rapport and develop synergy amongst its employees. Larry invites Alex and some other employees to form a circle of N people ($1 \leq N \leq 10$) in order to play Alex's favorite game, Puck Puck Moose. Some of the people in the circle do not get along, however, and Alex and Larry must accommodate their friends. Given the pairs of people who do not want to sit next to each other, determine the number of possible ways that the N people can sit down in the circle.

Note: A circle is rotationally symmetric, so any configuration that can be rotated into another is considered the same configuration.

SHORT NAME: `puck`

INPUT FORMAT:
The first line of input contains two integers, N and K ($1 \leq K \leq \binom{N}{2}$), where N is the number of people in the circle, numbered from 0 to $N-1$, and K is the number of pairs to follow. The next K lines consist of two integers, denoting the indices of the two people that do not want to sit next to each other.

Note: the order of the two indices does not matter. For example if 0 and 1 are a disallowed pair, 0 may not be neither to the left nor the right of 1.

OUTPUT FORMAT:
Output a single integer, the number of ways that the people can sit down in a circle, such that no pair that does not want to sit next to each other is together. If no configurations are possible, print 0.

SAMPLE INPUT:

```
4 2
0 1
2 3
```

SAMPLE OUTPUT:

```
2
```

F Candy Fest

TJ IOI Inc. is hosting its annual Candy Fest! Larry, who is organizing the event, has found a store that offers a massive sale on candy. Unfortunately, after purchasing a large amount of N different kinds of candy ($1 \leq N \leq 100,000$), he realized that the receipt of length M ($1 \leq M \leq 1,000$) has no spaces in it! Frustrated with this receipt format, Larry instead ate all of the candy himself! Help Larry figure out how much sugar Larry will consume after he eats all of the candy.

Note: It is guaranteed that no candy name is a prefix of another candy name and that a solution exists.

SHORT NAME: `candy`

INPUT FORMAT:
The first line of input contains the integer N, the number of different kinds of candy. The next N lines contain the name of the candy, a string of length L ($1 \leq L \leq 100$) given in all capital letters, followed by the amount of sugar in that candy a_i ($1 \leq a_i \leq 100,000$), separated by a space. The final line will contain the receipt, a string of capital letters.

OUTPUT FORMAT:
Output a single integer, the total amount of sugar that Larry will consume.

SAMPLE INPUT:

```
4
KITKAT 20
TWIX 28
REESES 8
SNICKERS 9
TWIXTWIXKITKATREESESTWIXSNICKERSTWIX
```

SAMPLE OUTPUT:

```
149
```

G Larryopoly

Tired of digging out change at the TJ IOI Inc. vending machines, Larry, being rich, decides to invent his own currency to be used by all TJ IOI employees. He creates N ($1 \leq N \leq 100$) different types of bills, each with a unique dollar value d_i ($1 \leq d_i \leq 1,000$).

Although many employees are initially skeptical, Niki decides to use this currency. Niki wants to feel rich, so he wants to maximize the number of bills he can hold. However, Niki refuses to take more than one bill of each kind. Niki will ask M times ($1 \leq M \leq 100,000$) if he can hold X amount of value in Larry's currency ($1 \leq X \leq 1,000,000$) and if so, how many bills that will take.

SHORT NAME: `larryopoly`

INPUT FORMAT:
The first line of input contains two integers N and M, where N is the number of bills and M is the number of queries. The next N lines each contain one integer, denoting the value of that bill, followed by M lines, each consisting of a query in the form of a value X.

OUTPUT FORMAT:
Output M lines, where each line contains the maximum number of bills that could make the value in the corresponding query, or -1 if this is impossible.

SAMPLE INPUT:

```
3 3
3
6
5
6
11
4
```

SAMPLE OUTPUT:

```
1
2
-1
```

H Cookie Baking

Devon, a dedicated member of the culinary staff at TJ IOI Inc., has made cookies for Kevin's birthday! However, smelling the aroma of chocolate chips coming from the kitchen, Alex devises a plot to bring cookies to him and his fellow employees.

Devon has N large piles of cookies ($1 \leq N \leq 100,000$) on the kitchen counter, where the i^{th} pile ($1 \leq i \leq N$) contains a_i cookies ($1 \leq a_i \leq 1,000,000$). Alex, on the other hand, wants to steal his cookies, but he brings along a different number of employees each time.

When he steals cookies, he wants to make sure he is able to split the cookies evenly among him and his $X-1$ employees (X people total). Alex always chooses X to be prime, because he likes prime numbers. Since the kitchen door is located next to cookie pile 1, Alex wants to take cookies from the first possible pile (i.e. the minimum value of i), such that he can split the pile's cookies amongst X people evenly.

Since Devon is a very efficient chef, whenever Alex takes cookies from a pile, he is able to restock the pile with exactly the same number of cookies. This means that the sizes of the cookie piles effectively do not change. Whenever Alex arrives, help him determine the best pile to take cookies from.

INPUT FORMAT:
The first line contains N and Q ($1 \leq Q \leq 100,000$). The second line contains N integers representing a_i. The next Q lines contains queries. Each of these lines consists of one integer X ($1 \leq X \leq 1,000,000$), where X is prime.

OUTPUT FORMAT:
For each query, output the minimum i such that X divides a_i, or output -1 if no such a_i exists.

SHORT NAME: `cookie`

SAMPLE INPUT:

```
5 6
2 15 49 11 17
3
7
2
3
13
5
```

SAMPLE OUTPUT:

```
2
3
1
2
-1
2
```

I Hungry Hungry Larrys

At the end of a long day at work, Devon stands in his cubicle at the northwest corner of an $N \times N$ grid ($1 \leq N \leq 800$) of cubicles on the 100^{th} floor of TJ IOI Inc. Each minute, Devon can move either one cubicle to the south or one cubicle to the east, and would like to reach the elevator, located in the cubicle at the southeast corner of the floor.

However, a number of Larrys $L_{i,j}$ reside in each cubicle ($0 \leq L_{i,j} \leq 9$), pretending to do work. Every time Devon moves into a new cubicle, each Larry in the cubicle that Devon moves into will reach into Devon's wallet and take one dollar. It is guaranteed that there are no Larrys in Devon's own cubicle (i.e., the one he begins on).

Thankfully, Devon's wallet has an infinite amount of money, but he still would like to lose as little money as possible. Determine the least amount of money that Devon must lose to the hungry hungry Larrys along the way, in order to reach the elevator and exit the building.

Note: North corresponds to up, south corresponds to down, east corresponds to right, and west corresponds to left.

SHORT NAME: `hungry`

INPUT FORMAT:
The first line of input contains the integer N, the size of the grid of cubicles. The next N lines each contain N integers, and together describe the number of Larrys within each cubicle.

OUTPUT FORMAT:
Output a single integer, the least amount of money that Devon must lose in order to get to the elevator.

SAMPLE INPUT:

```
4
0 2 5 1
2 9 3 0
4 6 1 2
8 2 2 6
```

SAMPLE OUTPUT:

```
16
```

J Grocery Shopping

On the opening day of the TJHSST Third Floor Grocery Shop, N ($1 \leq N \leq 100,000$) students arrive, where the ith student would like to buy A_i ($1 \leq A_i \leq 100$) items. The grocery shop has a total of M ($1 \leq M \leq 100,000$) checkout stations, and the number of seconds it takes for a student to checkout is exactly equal to the number of items that student purchases.

The grocery shop's management team has contracted TJ IOI Inc. to help ease some of the checkout congestion. The consultants at TJ IOI Inc. have devised an unusual plan to facilitate grocery checkouts in a calm (but inefficient) manner: all N students will line up in the order of arrival, and the management team will divide the students into M contiguous segments. Each segment will check out at a different checkout station, and all of the stations may operate at the same time. Please help the management team determine the minimum amount of time necessary to checkout all students if they divide the students optimally.

SHORT NAME: `grocery`

INPUT FORMAT:
The first line contains N and M. The next N lines contain an integer representing A_i.

OUTPUT FORMAT:
Output the minimum amount of time necessary to checkout all students.

SAMPLE INPUT:

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

SAMPLE OUTPUT:

```
6
```