Sixteenth Annual
University of Central Florida

High School Programming Tournament: Online Edition

Problems – Division 2

Problem Name	Filename
Alien Anatomy	anatomy
Bounding Bunny	bunny
Echoing Events	echo
Jetpack Escape	jetpack
Ranged Array	ranged
Sorting Gazebos	sorting
String Splitting	splitting
Sportsball Field	sportsball
Welcome!	welcome

Call your program file:
filename.cpp, *filename*.java, or *filename*.py

For example, if you are solving Bounding Bunny,
Call your program file:
bunny.cpp, bunny.java, or bunny.py
Call your Java class: bunny

Alien Anatomy

Filename: `anatomy`

While exploring the infinite cosmos, you have stumbled upon an alien species with bizarre anatomy. Similar to humans, their proteins are constructed of amino acids, but due to several mutations, they have 62 total amino acids, several of which perform the exact same function. To further understand their anatomy, you want to analyze their proteins.

You represent one of their proteins as a string of characters, where an alphanumeric character ('a'-'z', 'A'-'Z', or '0'-'9') represents an amino acid. You have found that the structure of a group of k adjacent amino acids determine the function of that group, but if one amino acid is substituted for an equivalent amino acid, then that group's function remains the same. A pair of amino acids is considered equivalent if the two amino acids are represented by different characters but perform the same function. Equivalency is symmetric, meaning that if amino acid a is equivalent to b, b is also equivalent to a.

In your analysis, you have now stumbled across the following problem: given a certain amino acid group of size k, determine how many distinct substrings exist in the protein string such that they perform an equivalent function. Two substrings of size k are considered distinct if at least one amino acid differs between the two substrings.

The Problem:

Given a protein string and the set of equivalencies between amino acids, for each query of amino acid groups, determine how many distinct substrings exist in the protein string that perform the same function as the query string.

The Input:

The first line of input will contain four integers, n, m, q and k ($1 \le n \le 10^5$; $1 \le m \le 30$; $1 \le q \le 500$; $1 \le k \le \min(n, 10)$), where n is the length of the protein string, m is the number of pairs of equivalent amino acids, q is the number of amino acid group queries, and k is the size of the groups. The next line will contain a string of length n composed of the characters 'a'-'z', 'A'- Z', and '0'-'9', representing the protein string.

The following m lines will consist of two space separated characters ('a' - 'z', 'A' - 'Z', or '0' - '9'), representing a pair of equivalent amino acids. It is guaranteed that each amino acid will appear in *at most* one equivalency pair, and no amino acid will be in an equivalency pair with itself. The last q lines of input will each contain a string of length k, corresponding to a query of an amino acid group, each formed of the same characters ('a'-'z', 'A'- Z', and '0'-'9').

The Output:

Output q lines, each consisting of a single integer: the number of distinct substrings that exist in the protein string that perform an equivalent function to the query group.

Sample Input 1:

```
12 4 2 4
HSPThsptHspT
H h
S s
t T
p P
hspt
TspH
```

Sample Output 1:

```
3
0
```

Sample Input 2:

```
7 2 3 3
0102010
0 1
2 3
000
102
131
```

Sample Output 2:

```
1
1
1
```

Bounding Bunny

Filename: `bunny`

Benny the Bunny is practicing his leaps! In order to do so, he built an obstacle course consisting of n blocks in a line. Since he is not very consistent, every time Benny jumps forward, he moves forward a uniformly distributed random integer number of blocks between 1 and n - 1 (inclusive).

If there is a chance he jumps beyond the end of the obstacle course, Benny will not jump forward. Instead, he will jump backwards exactly one block until he is sure that he will not land beyond the obstacle course if he jumped forward.

He's asked you to calculate how many leaps he should expect to take before he reaches the last block of the obstacle course from the first block.

The Problem:

Given the number of blocks in the obstacle course, calculate the expected number of jumps Benny will have to take to get from the first block to the last block in the obstacle course.

The Input:

The input consists of a single line containing one integer, n ($2 \leq n \leq 10^4$), representing the number of blocks in the obstacle course.

The Output:

Output one line containing a single real number l: the expected number of leaps Benny will have to take to get to the end of the obstacle course. Your answer will be considered correct if it is within an absolute or relative error of 10^{-4} of the judge's answer.

Sample Input 1:	**Sample Output 1:**
`2`	`1.00000`

Sample Input 2:	**Sample Output 2:**
`8`	`28.00000`

Echoing Events

Filename: `echo`

Today is just like any other day. You wake up, get ready, head to school, go to math class, take an exam, go to math class…wait, didn't that already happen? Something isn't right.

As you keep trying to go about your day, events keep reoccurring! You have named these time anomalies "event echoes" and now need to find a way to stop them! To find the root of what started these echoes, you need to identify the first event that repeated itself.

The Problem:

Given a list of events, determine the first event echo.

The Input:

The first line of input is a single line consisting of a single integer, n ($1 \leq n \leq 1{,}000$), representing the number of events that have occurred. The following n lines will consist of a single string containing at most 10 lowercase letters, representing the event that has occurred.

The Output:

Output the word representing the first event that repeated itself. The word should appear exactly the same as it was given in input, or "`NO ECHOES SPOTTED`" if no events repeat.

Sample Input 1:

```
3
awaken
dostuff
sleep
```

Sample Output 1:

```
NO ECHOES SPOTTED
```

Sample Input 2:

```
9
wakeup
getready
gotoschool
mathclass
test
mathclass
gohome
test
sleep
```

Sample Output 2:

```
mathclass
```

Jetpack Escape

Filename: `jetpack`

Growing bored with new games not meeting expectations, Vlad has decided to revisit a true classic: Jetpack Joyride. Vlad is a pro at gaming, so it is expected he won't struggle. However, since the game continues infinitely and increases in difficulty over time, it's inevitable that he will lose eventually. Enraged by this fact, Vlad would like to figure out how to move around one of the game's largest points of loss: zappers. For the moment he has decided to ignore all other aspects of the game other than the player and the zappers.

Zappers are electric fields that range in a vertical, horizontal, or diagonal line segment. When the player hits one of these zappers, they lose. Vlad wants to figure out if given his screen and the position he is in, he can make it to the end of the screen (i.e., the right side) without losing.

When playing, the player can only move in three ways:

- Move to the right only (balancing jetpack, or jetpack on the ground/roof)
- Move up and right if possible (jetpack on)
- Move down and right if possible (jetpack off)

The Problem:

Given the current state of the game, can you help Vlad figure out if he can make it through the zappers without losing?

The Input:

The first line of input contains two space separated integers, n and m ($2 \leq n \leq 10^4$; $2 \leq m \leq 10^4$; $4 \leq n \bullet m \leq 10^5$), representing the screen's height and width respectively. The next n lines will contain m characters, each representing a free space (`.`), a zapper (`z`), or the player (`b`). There will always be one and only one player.

The Output:

Output a single line whether it is possible for Vlad to get through the zappers. Output "`YES`" if he can get through them and "`NO`" if he cannot.

Sample Input 1:

```
4 10
b....z....
....z.....
....z....z
....z...z.
```

Sample Output 1:

```
YES
```

Sample Input 2:

```
4 10
b.....z...
....z.z...
....z.z..z
....z...z.
```

Sample Output 2:

```
NO
```

Sample Input 3:

```
10 20
....................
............zzzzzzz
......zz............
.....b.zz...........
........zz..........
.........zz.........
z.........zz........
z............z......
z............z......
............z......
```

Sample Output 3:

```
YES
```

Ranged Array

Filename: `ranged`

Thomas loves calling things he doesn't like *deranged*. He has decided he needs a term for the opposite phenomenon, so he's coined the term *ranged* to describe things that he enjoys. Thomas considers an array of integers *ranged* if, for every element in the array, the sum of all other elements is evenly divisible by that element.

Can you help Thomas determine whether a given array is ranged?

The Problem:

Given an array of integers, determine whether, for every integer in the array, the sum of all other elements is divisible by that integer.

The Input:

The input begins with a line containing a single integer, n ($2 \leq n \leq 5{,}000$), representing the size of the array. The second line of input consists of n integers, a_i ($1 \leq a_i \leq 10^5$), each representing the i^{th} element in the array.

The Output:

Output a single line. Output "`ranged`" if the array is ranged. Otherwise, output "`deranged`" if it is not.

Sample Input 1:

```
3
2 1 3
```

Sample Output 1:

```
ranged
```

Sample Input 2:

```
3
2 1 4
```

Sample Output 2:

```
deranged
```

Sorting Gazebos

Filename: `sorting`

The gazebo business is booming. What started as a curiosity about the word "gazebo" has turned into a multi-billion dollar corporation dedicated to selling gazebos. Thomas, the founder and CEO of this brilliant business, currently has n gazebo models in a line, each having a distinct ranking from 1 to n describing how fancy a given gazebo is (rank 1 is the fanciest). The only problem is the gazebos are not in the right order! Ideally, the gazebos would be sorted in order of fanciness (i.e. from 1 to n) as this will make for the most aesthetically pleasing gazebo lineup. This must be fixed immediately, but due to certain legal contracts Thomas signed with his company's marketing agency, he must use a particular method for rearranging the order of these gazebos.

Each day, Thomas will hire a contractor to reverse the order of some contiguous subsection of the gazebo lineup. Specifically, in a single operation, Thomas will choose two locations l and r in the lineup ($1 \leq l \leq r \leq n$) and reverse the order of the gazebos in the locations from l to r, inclusive. Since contractors are expensive, he would like to sort the gazebos in *at most* $2n$ days.

The Problem:

Given a permutation representing the fanciness of each gazebo, sort them into ascending order (i.e., most fancy to least fancy) using at most $2n$ range reversal operations.

The Input:

The first line of input will contain a single integer, n ($1 \leq n \leq 10^5$), representing the number of gazebos in the lineup. The following line will contain n unique integers, where the i^{th} integer, f_i ($1 \leq f_i \leq n$), represents the fanciness of the i^{th} gazebo.

The Output:

First, output a single line containing, d ($0 \leq d \leq 2n$), the number of days you will hire contractors. For each day, print a single line containing two integers, l and r ($1 \leq l \leq r \leq n$), representing the range of the gazebo lineup the contractors should reverse. After all operations are performed, the gazebo lineup must be sorted in increasing order. If there are multiple possible solutions, any valid solution according to the above constraints will be accepted. Note that you do not need to minimize the number of days used.

Sample Input 1:

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

Sample Output 1:

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

Sample Input 2:

```
2
2 1
```

Sample Output 2:

```
1
1 2
```

String Splitting

Filename: `splitting`

The Cat has just finished preparing his cake for Thing One and Thing Two's upcoming birthday! Now he has run into another conundrum: the Cat has a single string to divvy up for the two birthday Things. Thing One and Thing Two are very prone to jealousy, so he would like to make sure the strings that each of them receives are identical.

The Cat will first split the string into exactly two contiguous substrings (such that each letter of the original string is in exactly one of the two substrings). Then, he can choose to reverse one of the two strings. The Cat would like to determine if it is possible for him to make two identical strings from these operations.

The Problem:

Given a string, determine whether it is possible to split into two substrings that are identical, possibly after reversing one of the two strings.

The Input:

The input begins with a line containing a single integer, n ($2 \leq n \leq 10^5$), representing the length of the string. This is followed by a line consisting of a string of n lowercase English letters, representing the string the Cat would like to split.

The Output:

Output a single line. If it's possible for the Cat to split the string as needed, output "`Happy Birthday!`" Otherwise, output "`No split can fit!`"

Sample Input 1:

```
8
hspttpsh
```

Sample Output 1:

```
Happy Birthday!
```

Sample Input 2:

```
4
baba
```

Sample Output 2:

```
Happy Birthday!
```

Sample Input 3:

```
15
redfishbluefish
```

Sample Output 3:

```
No split can fit!
```

Sportsball Field

Filename: `sportsball`

Aria has settled into her new neighborhood and is now occupying her time inventing a new sport. Naturally, since Aria is such a big fan of triangles, she has devised a game, Sportsball, that is played on a triangular field.

The rules of Sportsball are far too complex to detail here, but Aria is sure that it will be a hit, so she has decided to construct the world's very first Sportsball field! For this, Aria has enlisted your help as her newly appointed Handler of Sportsball Plot Triangulation.

Aria has purchased a rectangular plot of land in the neighborhood, and now she needs you to find out the maximum possible area of a triangular Sportsball field that could be built within it.

The Problem:

Given the width and height of a rectangular plot of land, determine the maximum area of a triangle that can be built within the bounds of the plot.

The Input:

The input consists of one line containing two integers, w and h ($1 \le w \le 10^4$; $1 \le h \le 10^4$), representing the width and height of the rectangular plot, respectively.

The Output:

Output a single real number: the maximum area of a triangle that can be placed within the rectangular plot of land. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-2}.

Sample Input 1:	**Sample Output 1:**
`1 2`	`1.0000`

Sample Input 2:	**Sample Output 2:**
`3 3`	`4.5000`

Welcome!

Filename: `welcome`

Nic has created a new Discord server and invited his friends to it. Since his friends are a bunch of jokesters, every time someone joins, they wave hello to every single person who joined before them. Obviously, this can quickly get out of hand, but Nic would like your help with quantifying precisely how out of hand it can get. Can you help him?

The Problem:

Given a positive integer, n, denoting the number of people that joined the server, compute the total number of waves.

The Input:

The first and only line of input will consist of an integer, n ($1 \leq n \leq 100$), representing the number of people that joined the server.

The Output:

Output a single line consisting of the total number of waves.

Sample Input 1:	**Sample Output 1:**
1	0

Sample Input 2:	**Sample Output 2:**
3	3

Sample Input 3:	**Sample Output 3:**
5	10