Fourteenth Annual
University of Central Florida

High School Programming Tournament: Online Edition

Problems – Division 1

Problem Name	Filename
BABA Strings	baba
Truffula Bees	bees
Lior's Books	books
Cartesian Climbing	climb
Daniel's JUSTified Dilemma	just
Perilous Passage	passage
A Single Pepperoni	pepperoni
Surfing 101	surfing
What's Zupp?	zupp

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

For example, if you are solving Surfing 101,
Call your program file:
surfing.cpp, surfing.java, or surfing.py
Call your Java class: surfing

BABA Strings

Filename: `baba`

BABA is a cool sounding word! A "BABA" string is a string consisting of exclusively A's and B's such that the number of A's and B's in the string are equal. Given this property, note that a BABA string can have multiple substrings in it that are also BABA Strings. For example, for the string "BABA" the four BABA strings are BABA, BA, BA, and AB.

BABA the bunny would like to know if a given string is a BABA string. If so, BABA the baby bunny would also like to know how many substrings of that BABA string are also BABA strings. BABA, being a bunny and therefore not understanding math, the English language or the concept of a "substring", is struggling to do this. Would you mind writing a program to help them?

ABBA, the brother of BABA, would like you to know that a substring is defined as non-empty continuous subsequence of characters.

The Problem:

Given a string, determine whether it is a BABA string, and if so, determine how many substrings of the given string are also BABA strings.

The Input:

The input is a single line containing a string, *s*, of length between 1 and 100,000 (inclusive). The string will contain only uppercase A's or B's.

The Output:

Output a single line containing a single integer representing the number of substrings of the given string that are BABA strings. If the inital input string is not a BABA string at all, output "`BABA IS NOT YOU!`" instead (even though these strings may have substrings that are BABA strings,we will ignore that substrings here).

Sample Input 1:	Sample Output 1:
`BABA`	`4`

Sample Input 2:	Sample Output 2:
`BABBB`	`BABA IS NOT YOU!`

Sample Input 3:	Sample Output 3:
`AAABBB`	`3`

Truffula Bees

Filename: `bees`

But those bees! Those bees!

Those Truffula Bees!

All my life I'd been searching

for bees such as these.

- Adapted from Dr. Seuss's *The Lorax*

The Truffula Bees are flourishing in Truffula Forest under their good and fair ruler, Queen Bee. Their idyllic paradise charms any creature who enters.

Each bee living in Truffula Forest has a favorite number. A group of bees is considered to be a *best friend group* if for every pair of bees in the group, their favorite numbers multiply to form a perfect square. In particular, note that any group consisting of only one bee is a best friend group.

To promote camaraderie among the bees in her queendom, Queen Bee wishes to partition all of the bees into best friend groups such that each bee is part of exactly one group, and the number of groups is as small as possible.

For example, if we have nine bees with favorite numbers 1 to 9, respectively. These bees may be partitioned as follows: [1, 4, 9], [2, 8], [3], [5], [6], [7]. Bees 1, 4 and 9 may be grouped together because the product of each pair of elements in this group is equal to a perfect square ($1 \times 4 = 4$, which is a perfect square; $1 \times 9 = 9$, which is a perfect square; and $4 \times 9 = 36$, which is also a perfect square). Likewise, bees 2 and 8 may be grouped together (2 x 8 = 16, which is a perfect square). The remaining bees cannot be paired, so they are each in groups of their own. The gives a total of six groups, and it can be shown that this is the smallest possible number of groups.

As another example, if we have two bees with favorite numbers 10 and 20, they cannot be paired (10 x 20 = 200, which is *not* a perfect square). Therefore, each bee is in a group by itself, for a total of two groups.

The Problem:

Determine the smallest number of best friend groups required to partition all of the bees in Truffula Forest.

The Input:

The first line of input contains an integer, n ($1 \le n \le 10^5$), representing the number of bees. The second line of input contains n integers, a_i ($1 \le a_i \le 10^4$), representing the favorite number of each bee.

The Output:

Output a single integer: the smallest number of best friend groups.

Sample Input 1:	Sample Output 1:
9 1 2 3 4 5 6 7 8 9	6

Sample Input 2:	Sample Output 2:
5 1 4 9 16 25	1

Sample Input 3:	Sample Output 3:
2 10 20	2

Lior's Books

Filename: `books`

Lior has become quite the avid reader and has assembled an impressive collection of books. He has stacked these books on a table to varying heights and eventually plans to organize them. For now, Lior will settle with all the book stacks having the same number of books. Since Lior would like to go back to playing Super Auto Pets, he wants to make as few moves as possible. A move consists of taking the top book from any stack and placing it on top of any other stack. Lior has asked you to write a program that will determine whether or not he can make all book stacks the same height. If it is possible, he would also like to know the minimum number of movements necessary so he can complete this tedious task and go back to playing Super Auto Pets as quickly as possible!

The Problem:

Given a list of how many books are in each stack, determine if it is possible to make all of the stacks the same height and if so, what is the minimum number of movements that must be made to make all of the stacks equal height.

The Input:

The input will start with a line containing a single integer, s ($1 \leq s \leq 10^3$) representing the number of book stacks Lior currently has. The following line will contain s integers, h_i ($0 \leq h_i \leq 10^9$), representing the height of the ith stack, respectively.

The Output:

Output a single integer: the minimum number of movements Lior must complete to make all of the stacks have an equal number of books. If it is impossible to do so, output "impossible" instead.

Sample Input 1:	Sample Output 1:
`4` `1 7 5 4`	`impossible`

Sample Input 2:	Sample Output 2:
`2` `1 5`	`2`

Sample Input 3:	Sample Output 3:
`1` `5`	`0`

Cartesian Climbing

Filename: `climb`

Jacob has decided to go indoor rock climbing at the University of Central Florida's climbing tower! Routes on the climbing tower are defined by a set of holds, represented as points in the Cartesian plane, onto which a climber can grab. A route will begin at the hold with the lowest height and will end at the hold with the highest height, and it is always guaranteed that the ending hold will be higher than the starting hold.

Jacob has a unique (yet impressive) climbing technique that only makes uses of his arms. Therefore his position can be defined entirely by the hold to which he is currently attached. He can only traverse from hold to hold if the Euclidean distance between his current hold and his target hold is less than or equal to his arm length. Due to an experiment involving his roommate, Tyler, three extra cheesy pizzas, and a high-powered laser, Jacob has gained the ability to grow and shrink the length of his arms at will (albeit, painfully) and would like to know what the minimum arm length needed to complete the route. Note that Jacob does not need to visit every hold in a route in order to complete it; he only needs to reach the highest hold in the route.

The Problem:

Given a list of holds as points in the Cartesian plane that represents the climbing tower Jacob plans on traversing, determine the minimal arm length needed to complete the given route.

The Input:

The input will begin with a line containing a single integer, h ($2 \leq h \leq 500$), representing the number of holds on a given tower. Following this, each of the next h lines will consist of two integers, x ($0 \leq x \leq 1{,}000$) and y ($0 \leq y \leq 1{,}000$), representing the location of a hold on the tower. No two holds will occur at the same place and no hold will have the same y value as the starting or ending holds.

The Output:

For the given tower, output a line containing a single number representing the minimum arm length needed to complete the given tower (i.e., get from the starting hold to the ending hold). Any answer within 10^{-6} absolute or relative error of the answer will be considered correct.

(Sample Inputs and Sample Outputs appear on the following page)

Sample Input 1:	**Sample Output 1:**
4 5 1 3 3 4 0 4 5	2.828427124

Sample Input 2:	**Sample Output 2:**
5 5 1 4 0 7 3 3 3 6 6	3.162277660

Daniel's JUSTified Dilemma

Filename: `just`

Daniel really just despises the word "just." Of all the words in the English language, he thinks "just" is just extremely toxic and will call out anyone for using this just awfully horrific word. In fact, he despises any word that implies a problem is easier than it really is. Therefore, he views the words "simple" and "trivial" to be extremely toxic as well.

What he doesn't realize is that he is always using these toxic words, just in a more discrete manner. He tends to have these words embedded as a subsequence in his sentences. A subsequence of text is defined as a sequence obtained by deleting some characters (possibly none) from the text without changing the order of any characters. For instance, "hspt" is a subsequence of the text "hospital" but not a subsequence of "photonics" or "hosting." The toxicity is the largest integer *k* such that the concatenation of any *k* "toxic words" appears as a subsequence in the sentence.

For instance, in the sentence "the justified justice on jupiters moon thebe," "just" can be combined three times as "justjustjust" and is a valid subsequence of the sentence, so the sentence has a toxicity of 3. Similarly, in the sentence "simon says please jump and sing take on me," the words "simple" and "just" can be combined as "simplejust" and is a valid subsequence of the sentence, so this sentence has a toxicity of 2. Daniel just cannot stand being toxic and wants to know the toxicity of his sentences.

The Problem:

Given Daniel's sentence, determine its toxicity.

The Input:

Each sentence will begin with a single integer, *n* ($1 \le n \le 2{,}000$). The next line will be the sentence that Daniel wants analyzed, which will be composed of between 1 and 2,000 characters, inclusive. This sentence will only contain lowercase letters and spaces. Following will be *n* lines, each containing a single word representing one of the words that Daniel views as toxic. This word will only contain lowercase letters and be of at least 1 in length. The sum of the lengths of all *n* words will be less than 2,000.

The Output:

For each sentence, output a single integer: the toxicity of Daniel's sentence.

(Sample Inputs and Sample Outputs appear on the following page)

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

1 justified justice on jupiters moon thebe just	3

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

2 simon says please jump and sing take on me just simple	2

Perilous Passage

Filename: `passage`

Oliver is the captain of a fleet of ships which he is guiding through the Mediterranean Sea. While sailing through the archipelago of Greece, Oliver traveled to the mysterious island of Delphi to receive a prophecy from the Oracle of Delphi. The Oracle warned Oliver of a perilous passage guarded by two sea monsters: the ravenous Scylla and the chaotic Charybdis. These two creatures were situated opposite one another along a passage through which Oliver needed to guide his fleet. If one of his ships strayed too far to the right, it could be ripped apart by one of Scylla's many wolf-headed appendages. If one of them strayed too far to the left, then it could be decimated by Charybdis' volatile whirlpool.

Luckily, Oliver was able to gather intel about when Scylla and Charybdis would awaken. Fortunately for Oliver as well, Scylla and Charybdis are only grumpy at the exact moment they wake, so they will only destroy ships that are within r units of them right as they wake up. He has modeled this passage as a set of two vertical lines in the Cartesian plane that are d meters apart. Scylla is situated along the left vertical line at $x = 0$ and Charybdis is situated along the right vertical line at $x = d$. Scylla and Charybdis are both situated on the x-axis. The ships are narrow so they can be modeled as vertical line segments that sail vertically. A ship is only in danger of being destroyed by Scylla or Charybdis if some point of the ship is on the x-axis at the exact moment the monsters wake.

Oliver plans to send each ship through the passage at a certain time and speed, and at a certain x-coordinate. Because his fleet contains a lot of ships, he has asked you to determine how many ships will make it through the passage under his current plan.

The Problem:

Given the times Scylla and Charybdis awaken, how far they will reach to destroy ships, and the information for each ship (the x-coordinate of the ship, the time the ship enters the range, and the time the ship exits the range), determine how many ships will make it through the passage.

The Input:

Input will begin with a line containing three integers, d, n and a ($2 \le d \le 10^5$; $1 \le n \le 10^5$; $1 \le a \le 10^5$), where d is the distance between the two sides of the passage, n is the number of ships that travel through the passage, and a is the number of times Scylla and Charybidis awaken. The following n lines will contain three integers, x, s and e ($0 < x < d$; $1 \le s \le 10^5$; $1 \le e \le 10^5$), where x is the x-coordinate of where the ship crosses the x-axis, s is the time that the ship enters the range, and e is the time that the ship exits the range.

After that, a lines will follow each containing two integers, z and r ($0 \le r \le d$; $1 \le z \le 10^5$) where z is the time Scylla and Charybdis awaken, and r is the maximum distance away from them that Scylla and Charybdis will attack, respectively. All ships on the x-axis within r units of Scylla or Charybdis will be destroyed. In Oliver's plan, ships will only share the same x-coordinate as long as they never cross the x-axis at the same time (otherwise the two ships will crash!).

The Output:

Given the passage, output a single line containing a single integer: the number of ships that safely make it through the passage.

Sample Input 1:	Sample Output 1:
10 3 1 3 2 7 8 4 10 5 1 8 8 4	2

Sample Input 2:	Sample Output 2:
10 3 2 3 2 6 8 4 10 5 1 7 4 3 12 5	1

A Single Pepperoni

Filename: `pepperoni`

Jacob is known for his legendary pizza-making skills. He makes only the most perfectly circular pizzas covered with robust tomato sauce, thick-grated mozzarella cheese, fresh-grown basil, and, of course, large pepperoni. Today is Tyler's favorite day: pizza day! Jacob made a delicious pepperoni pizza, as always, except for one problem: someone ate all his pepperoni! Well… not all of it. Standing alone in the bag was a single, beautiful, gorgeous, and, most importantly, circular piece of pepperoni. Thank heavens! Jacob put together a wonderful "*one-topping*" pizza.

Fortunately, Tyler loves cheese! Jacob threw the pizza in the oven and left to go purchase more ingredients for next week, and so Tyler is in charge of cutting the pizza. Jacob is extremely meticulous about pizza-making, so he knows the exact radius of the pizza as well as the location and radius of the pepperoni. Tyler doesn't like pepperoni, so out of curiosity, he wonders the minimum area of a pizza slice that would contain the pepperoni entirely without cutting through the pepperoni. As a reminder, a slice of pizza is the sector of the pizza that is bounded by two rays starting at the center of the pizza.

The Problem:

Given the measurements of the pizza, determine the minimum area of a pizza slice that would contain the single pepperoni.

The Input:

The input will be a single line containing four positive integers, r, x, y, and p ($r < 1,000$; $x < 1,000$; $y < 1,000$; $p < 1,000$), representing the radius of the pizza, the relative x and y coordinate of the pepperoni to the center of the pizza, and the radius of the pepperoni, respectively. The pepperoni will not intersect or contain the center or crust (edge) of the pizza (because that would be ridiculous!).

The Output:

Output a single number, a, the minimum area of the pizza slice that entirely contains the single pepperoni. Your answer will be considered correct if it is within an absolute or relative error of 10^{-6}.

Sample Input 1:	Sample Output 1:
`10 1 6 2`	`33.50305004`

Sample Input 2:	Sample Output 2:
`5 2 3 1`	`7.025872538`

Surfing 101

Filename: `surfing`

Welcome to Loopdeeloop Planet! I'm your surfing coach, here to get you set up on the track.

We are proud to own the only ray surfing racetrack in the universe. As you can see, it's a loop of wavy water a million miles in the sky. Our track is $2n$ meters wide; each edge of the track is located n meters from the center. If, at any point, your horizontal position on the track falls below $-n$ or above n, you will fall.

Watch out! The waves are rough, and often unpredictable. During your run, a total of w waves will come towards you. Each wave has the potential to displace you by some horizontal amount: we know with certainty that wave i will displace you anywhere from a_i to b_i meters. Formally, wave i will add a real number, x, to your horizontal position,such that $a_i \leq x \leq b_i$.

Safety is one of our top ten priorities here on Loopdeeloop Planet, and we never wish for our surfers to fall. I'll let you choose your starting horizontal position on the track, and we'll see where you end up!

The Problem:

Given the bounds of the track and the information for each wave, determine whether there exists some horizontal position on the track such that you are guaranteed not to fall if you start there.

The Input:

The first line of input contains two integers, w ($1 \leq w \leq 10^5$) and n ($1 \leq n \leq 10^9$), representing the number of waves and the positions of the track boundaries, respectively. The following w lines each contain two integers, a_i and b_i ($-n \leq a_i \leq b_i \leq n$), indicating that wave i could displace you anywhere from a_i to b_i meters, inclusive.

The Output:

If there exists some horizontal position on the track such that you are guaranteed not to fall if you start there, print "`Hang ten, dude!`" Otherwise, print "`Free fallin'`" instead.

(Sample Inputs and Sample Outputs appear on the following page)

Sample Input 1:

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

Sample Output 1:

```
Hang ten, dude!
```

Sample Input 1:

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

Sample Output 1:

```
Free fallin'
```

Sample Input 3:

```
1 100
-100 100
```

Sample Output 3:

```
Hang ten, dude!
```

What's Zupp?

Filename: `zupp`

Jacob, Tyler, and Daniel just downloaded this great app for students called Zupp (this is not sponsored, we just like the app). Zupp is an app that gives you free food from various locations, many of which are pretty good! It does have one shortcoming, though: it doesn't have an Android build. This is fine for Jacob and Tyler, as they have iPhones, but Daniel just got a sweet new Google Pixel, so he can't download Zupp.

This is no problem, though. Even though Daniel can't download the app, he can still make an account and use Tyler's phone to login and claim the deals (Jacob's phone has chronically low battery and so cannot be reliably trusted to switch users). It takes c minutes to change between Zupp accounts on Tyler's phone.

Tyler and Daniel visit a very long, line-like shopping plaza with a number of shops having free food on Zupp (while Jacob goes along to watch). Naturally, they want to visit them all. Due to the shape of the shopping plaza, they can represent all the shops as laying on an infinitely long line, and each shop at a distinct non-negative x-coordinate. It takes 1 minute to travel 1 unit on this axis. They can start and end at any coordinate on the line, and must all stay together.

For example, if it takes 5 minutes to switch accounts on the phone, and the locations are at 0 and 10, then the optimal strategy is to start at x=0, switch users, then travel to x=10 and switch users once more. This results in everyone claiming every deal in a total of 20 minutes. However, if we change the amount of time to switch accounts to be 50 minutes, then it is optimal to start at x=0, travel to x=10, switch users, and then travel back to x=0. This results in everyone claiming every deal in a total of 70 minutes.

What is the minimum amount of time it would take for Tyler and Daniel to claim the deal at every shop? Note that in order for someone to claim a deal at a shop, they must be logged into a phone at that shop. Assume that claiming a deal is instantaneous. As a reminder, Tyler and Daniel must share a phone.

The Problem:

Given the number of minutes it takes to switch users, and the locations of the shops Tyler and Daniel (along with Jacob) want to visit, what is the minimum amount of time it takes them to claim every deal at every location?

The Input:

The first line of input will contain two integers, n and c ($1 \leq n \leq 2{,}000$; $0 \leq c \leq 10^5$), representing the number of shops and the number of minutes it takes to switch users, respectively. Then, a line containing n integers will follow, x_i ($0 \leq x_i \leq 10^5$), representing the locations of the n shops in sorted order, respectively.

The Output:

Output a single integer: the minimum amount of minutes it would take for everyone to claim every deal at every shop.

Sample Input 1:	**Sample Output 1:**
2 5 0 10	20

Sample Input 2:	**Sample Output 2:**
2 50 0 10	70