International Collegiate Programming Contest

2020

Latin American Regional Contests

July 3, 2021

Warmup Session

This problem set contains 3 problems; pages are numbered from 1 to 6.

This problem set is used in simultaneous contests hosted in the following countries:

Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador, El Salvador, Guatemala, Jamaica, México, Nicaragua, Perú, Puerto Rico, República Dominicana, Trinidad y Tobago and Venezuela

v1.0

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp`, *codename*`.java`, *codename*`.kt`, *codename*`.py3`, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

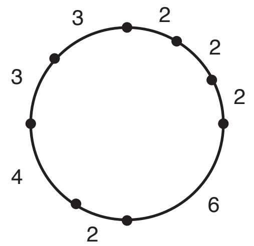
Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.
6. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test case is such that there are no ties when rounding as specified.

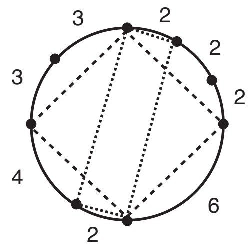
Problem A – Building a Field

John is a meticulous person. In his farm he built a circular field with some trees planted right at the circumference of the field. Figure (a) below shows the field with the trees.

Now John wants to use a long rope and four of the field trees to demarcate a rectangle using the trees as vertices and the rope as edges. Figure (b) below shows two rectangles that can be demarcated using the trees of the field in figure (a).



(a)



(b)

Given the description of the positions of the trees in John's circular field, you must determine whether it is possible to demarcate a rectangle using four of the trees as vertices and the rope as edges.

Input

The first line contains an integer N ($4 \leq N \leq 10^5$) indicating the number of trees in the field. Trees are represented as points on a circumference. The second line contains N integers $L_1, L_2, \ldots, L_N$ ($1 \leq L_i \leq 10^6$ for $i = 1, 2, \ldots, N$) indicating the arc lengths between each pair of consecutive trees. The arcs are given in counter-clockwise order. The total length of the circumference does not exceed 10^9.

Output

Output a single line with the uppercase letter "`Y`" if it is possible to demarcate a rectangle using the given trees, and the uppercase letter "`N`" otherwise.

Sample input 1	**Sample output 1**
`8` `3 3 4 2 6 2 2 2`	`Y`

Sample input 2	**Sample output 2**
`4` `14 16 15 15`	`N`

Sample input 3	**Sample output 3**
`6` `3 7 7 3 10 10`	`Y`

This page would be intentionally left blank if we would not wish to inform about that.

Problem B – Database of Clients

Nowadays there are billions of email users. A little-known fact is that some email providers offer way more than the usual `username@provider.com` email address.

Some providers simply ignore dots in usernames. Thus, if John owns the username `johnsmith`, he could tell people that his email address is `johnsmith@provider.com`, `john.smith@provider.com` or `john.s.mith@provider.com`, among others. Emails sent to any of these addresses would end up on his mailbox.

Other providers allow appending the character "`+`" followed by any combination of letters and/or digits after the username. With this feature, by registering the username `johnsmith`, John would also be able to use `johnsmith+friends@provider.com` and `johnsmith+2x3is6@provider.com`.

Sometimes both features are available at once and in those cases `john.smith+icpc@provider.com` and `john.smith+wants.2.eat.lemon.3.14@provider.com` are valid addresses that John could use.

This is quite useful for users, who can manage different addresses to help organize their mailboxes and easily filter the newsletters eventually sent after registering on a new website. Unfortunately, this also opens up space for abuse. Some websites rely upon the fact that each email address identifies a single user. However, a misbehaving user might easily create multiple accounts by taking advantage of the multiple addresses allowed by the email provider.

After learning all of this your boss got really worried. What if the number of unique users that has been reported to the shareholders is not accurate, bloated by duplicate accounts instead? That brings you to the task at hand: given the list of all email addresses from the users database of the company, you must determine the real number of unique users, assuming that all email providers have both described features available.

Input

The first line contains an integer N ($1 \leq N \leq 1000$) representing the number of email addresses in the database. Each of the next N lines contains a string of at most 100 characters representing an email address in the database. Each email address has the form *localpart*`@`*provider* where *localpart* is a non-empty list of labels with a "`.`" (dot) or a "`+`" (plus sign) between each pair of consecutive labels, and *provider* is a non-empty list of labels always with a "`.`" (dot) between each pair of consecutive labels. A label is a non-empty sequence of lowercase letters and/or digits. The character "`+`" (plus sign) appears at most once in each email address.

Output

Output a single line with an integer indicating the number of unique users in the database.

Sample input 1	Sample output 1
2 two.different.providers@now.here two.different.providers@nowhere	2

Sample input 2	Sample output 2
2 1.2.3@testing testing@1.2.3	2

Sample input 3	Sample output 3
7 alice@e.mail eve@another.mail bob@e.mail joe90@e.mail b.o.b@e.mail bob+new@e.mail bob@another.provider	5

This page would be intentionally left blank if we would not wish to inform about that.

Problem C – Mountain Ranges

Famous for its mountain ranges, Nlogonia attracts millions of tourists every year. The government has a dedicated budget for continuous maintenance of the hiking trails spread across the country and most of them are filled with scenic viewpoints, accessible through wooden walkways and stairs.

Currently on a trip through Nlogonia and with hopes of going back home with lots of breath-taking pictures, Lola and her husband want to visit as many viewpoints as possible. They plan to hike a different trail each day and explore its viewpoints. However, to avoid being exhausted at the end of the day, if moving from one viewpoint to the next requires going up more than X meters they simply call it a day and go back to their hotel in order to get some rest. Fortunately, every hiking trail in Nlogonia is equipped with modern chairlifts, so the couple can start hiking the trail at any viewpoint they decide. Once the hiking begins the couple only moves towards the peak of the mountain.

To make sure she doesn't waste a day Lola only wants to hike on trails where she'll get to a reasonable number of viewpoints. Given the altitudes of the scenic viewpoints on a hiking trail, you must determine the maximum number of viewpoints that the couple can visit.

Input

The first line contains two integers N ($1 \leq N \leq 1000$) and X ($0 \leq X \leq 8848$), indicating respectively the number of scenic viewpoints on the hiking trail, and the maximum number of meters that Lola and her husband are willing to go up from one viewpoint to the next. The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 8848$ for $i = 1, 2, \ldots, N$), where A_i is the altitude (in meters) of the i-th viewpoint. Viewpoints are given in the order they appear on the hiking trail and their altitudes are non-decreasing, that is, $A_i \leq A_{i+1}$ for $i = 1, 2, \ldots, N - 1$.

Output

Output a single line with an integer indicating the maximum number of scenic viewpoints that can be visited without going up more than X meters from one viewpoint to the next, and considering that the journey can be started at any viewpoint.

Sample input 1	Sample output 1
`9 2` `3 14 15 92 653 5897 5897 5898 5900`	`4`

Sample input 2	Sample output 2
`9 0` `3 14 15 92 653 5897 5897 5898 5900`	`2`

Sample input 3	Sample output 3
`9 8848` `3 14 15 92 653 5897 5897 5898 5900`	`9`

This page would be intentionally left blank if we would not wish to inform about that.