**Largest subarray with GCD one**

Show Topic Tags   

Given an array with n elements. Find length of the largest sub-array having GCD equal to 1. If no such  subarray exist with GCD 1, print -1.

Examples:

Input : 1 3 5

Output : 3

Input : 2 4 6

Output :-1

**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer n denoting size of the array. Then in the next line are n space separated values of the array.  
  
**Output:**  
For each test case in a new line print the required output.  
  
**Constraints:**  
1<=T<=100  
1<=n<=100  
1<=A[]<=100  
  
**Example:  
Input:**  
2  
3  
2 4 6  
3  
1 3 5  
  
**Output:**  
-1  
3

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/largest-subarray-with-gcd-one/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication251;

import java.io.\*;

import java.math.\*;

import java.util.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication251 {

static int Gcd(int a, int b)

{

if (a == 0)

return b;

return Gcd(b%a, a);

}

static int findLargest(int arr[], int n)

{

/\*If gcd of any subarray is 1 then gcd of

any number with the sub array will be 1.

so if we are getting any subarray with

gcd 1, then maximum number of element of

the subarray will be equal to the number

of elements of the array. Else it will be -1.\*/

int gcd = arr[0];

for (int i=1; i<n; i++)

gcd = Gcd(gcd, arr[i]);

return (gcd == 1)? n : -1;

}

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String[] input = br.readLine().trim().split(" ");

int[] a = new int[n];

for(int i =0; i<n; i++) {

a[i] = Integer.parseInt(input[i]);

}

System.out.println( findLargest(a,n));

}

}

}

-----------EDITORIAL------------------

Largest subarray with GCD one

There is an array with n elements. Find length of the largest subarray having GCD equal to 1. If no subarray with GCD 1, then print -1.

Examples:

Input : 1 3 5

Output : 3

Input : 2 4 6

Output :-1

[**Recommended: Please solve it on “*PRACTICE*” first, before moving on to the solution.**](http://practice.geeksforgeeks.org/problems/largest-subarray-with-gcd-one/0)

A **simple solution**is to consider every subarray and find its GCD and keep track of largest subarray with GCD one. Finally return length of the largest subarray with GCD 1.

An **efficient solution** is based on fact that if any two elements have GCD equals to one, then whole array has GCD one. So the output is either -1 or length of array.

|  |
| --- |
| // C++ progra, to find length of the largest  // subarray with GCD equals to 1.  #include<bits/stdc++.h>  using namespace std;    int findLargest(int arr[], int n)  {      /\*If gcd of any subarray is 1 then gcd of       any number with the sub array will be 1.       so if we are getting any subarray with       gcd 1, then maximum number of element of        the subarray will be equal to the number        of elements of the array. Else it will be -1.\*/      int gcd = arr[0];      for (int i=1; i<n; i++)          gcd = \_\_gcd(gcd, arr[i]);        return (gcd == 1)? n : -1;  }    // Driver code  int main()  {      int arr[] = {1, 3, 5, 7};      int n = sizeof(arr)/sizeof(int);      cout << "Length of the largest subarray = "           << findLargest(arr, n);      return 0;  } |

Run on IDE

Output:

Length of the largest subarray = 4