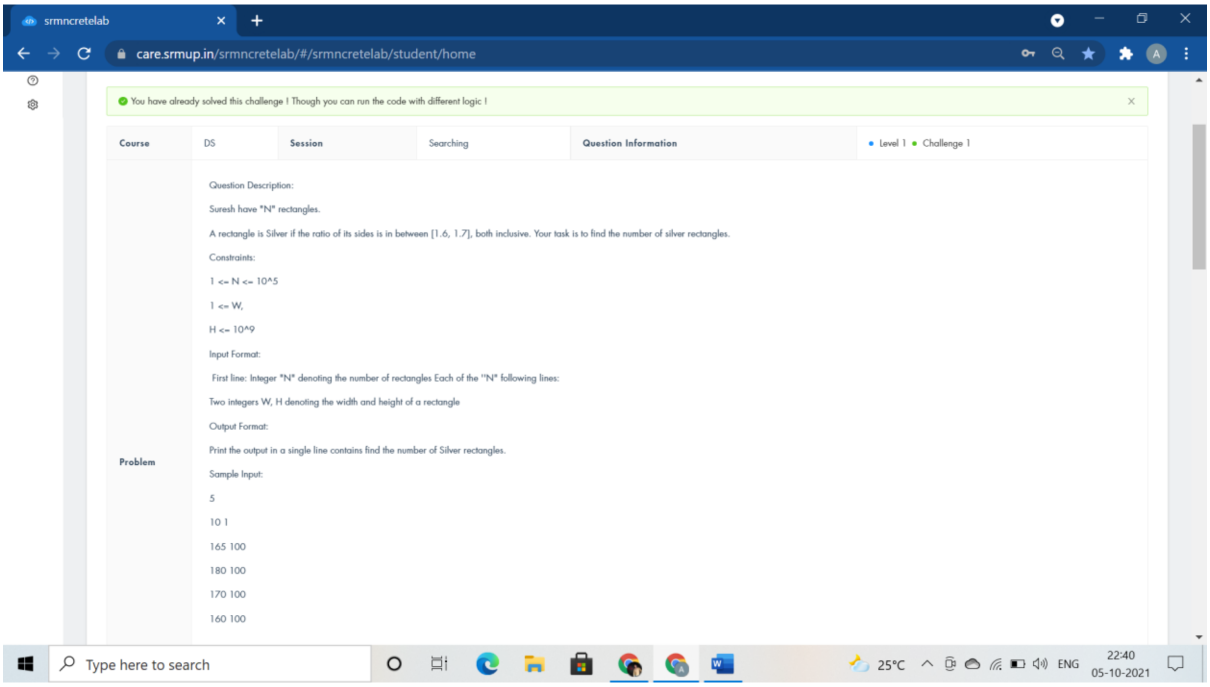
DSA SEARCHING



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	• Level 1 • Challenge 1

Problem

Question Description:

Suresh have "N" rectangles.

A rectangle is Silver if the ratio of its sides is in between [1.6, 1.7], both inclusive. Your task is to find the number of silver rectangles.

Constraints:

1 <= N <= 10^5

1 <= W,

H <= 10^9

Input Format:

First line: Integer "N" denoting the number of rectangles Each of the ''N" following lines:

Two integers W, H denoting the width and height of a rectangle

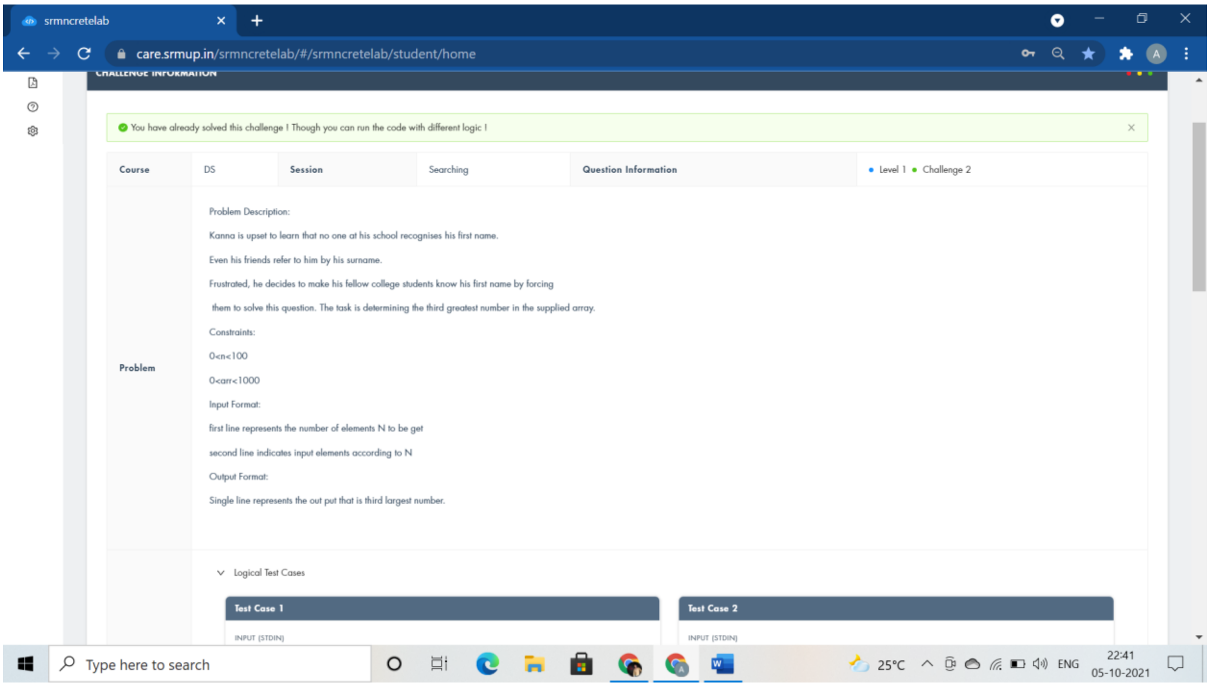
Output Format:

Print the output in a single line contains find the number of Silver rectangles.

Sample Input:

```
5
10 1
165 100
180 100
170 100
160 100
```

```
#include <stdio.h>
#include<math.h>
int main()
{
   float n,i,width,height;
   scanf("%f",&n);
   int count=0;
   for(i=0;i<n;i++)
   {
      scanf("%f %f",&width,&height);
      if(width/height>=1.6 && width/height<=1.7)
      ++count;
      else if(height/width >=1.6 && height/width<=1.7)
      ++count;
   }
   printf("%d",count+1);
         return 0;
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	● Level 1 ● Challenge 2

Problem

Problem Description:

Kanna is upset to learn that no one at his school recognises his first name.

Even his friends refer to him by his surname.

Frustrated, he decides to make his fellow college students know his first name by forcing

them to solve this question. The task is determining the third greatest number in the supplied array.

Constraints:

0<n<100

0<arr<1000

Input Format:

first line represents the number of elements N to be get

second line indicates input elements according to N

Output Format:

Single line represents the out put that is third largest number.

Logical Test Cases

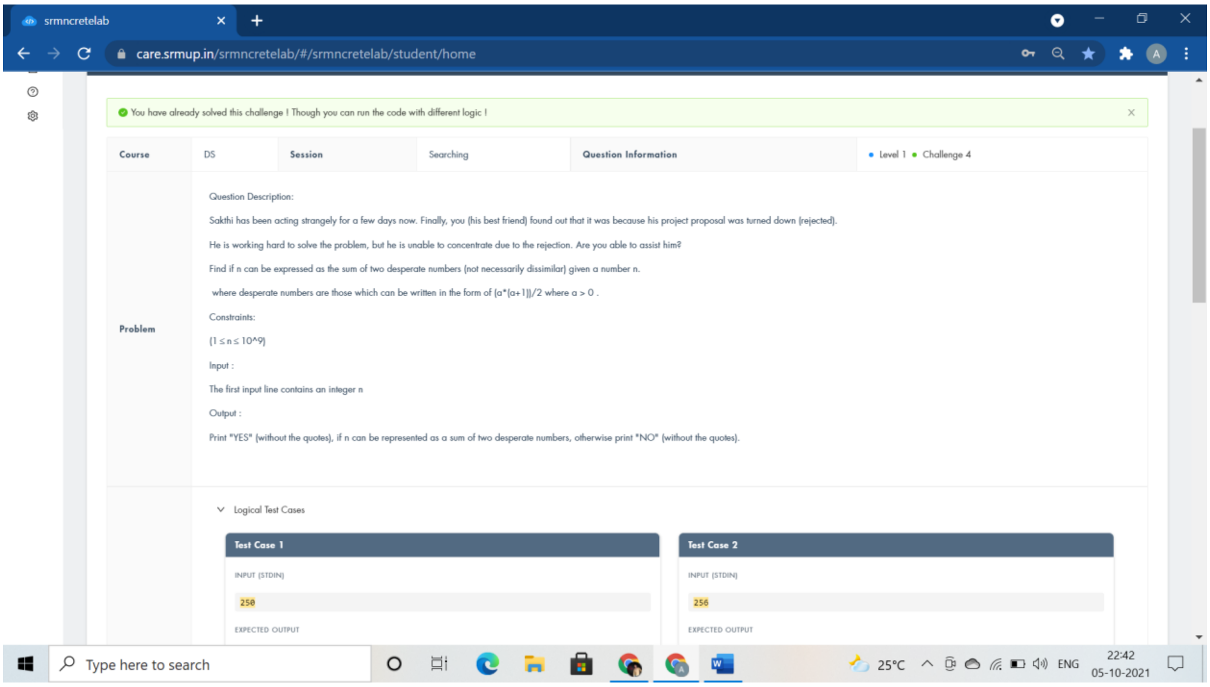
Test Case 1

INPUT (STDIN)

Test Case 2

INPUT (STDIN)

```
#include <stdio.h>
void thirdLargest(int arr[],int arr_size)
{
  int j,k,temp;
  for(j=0;j<arr_size;j++)
  {
    for(k=j+1;k<arr_size;k++)
    {
      if(arr[j]>arr[k])
      {
      temp=arr[j];
      arr[j]=arr[k];
      arr[k]=temp;
    }
    }
  }
}
int main()
{
  int i,n;
  scanf("%d",&n);
  int arr[n];
  for(i=0;i<n;i++)
  scanf("%d",&arr[i]);
  thirdLargest(arr,n);
  printf("The third Largest element is %d",arr[n-3]);
          return 0;
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	• Level 1 • Challenge 4

Problem

Question Description:

Sakthi has been acting strangely for a few days now. Finally, you (his best friend) found out that it was because his project proposal was turned down (rejected).

He is working hard to solve the problem, but he is unable to concentrate due to the rejection. Are you able to assist him?

Find if n can be expressed as the sum of two desperate numbers (not necessarily dissimilar) given a number n.

where desperate numbers are those which can be written in the form of (a*(a+1))/2 where a > 0 .

Constraints:

(1 ≤ n ≤ 10^9)

Input :

The first input line contains an integer n

Output :

Print "YES" (without the quotes), if n can be represented as a sum of two desperate numbers, otherwise print "NO" (without the quotes).

Logical Test Cases

Test Case 1

INPUT (STDIN)

```
250
```

EXPECTED OUTPUT

Test Case 2

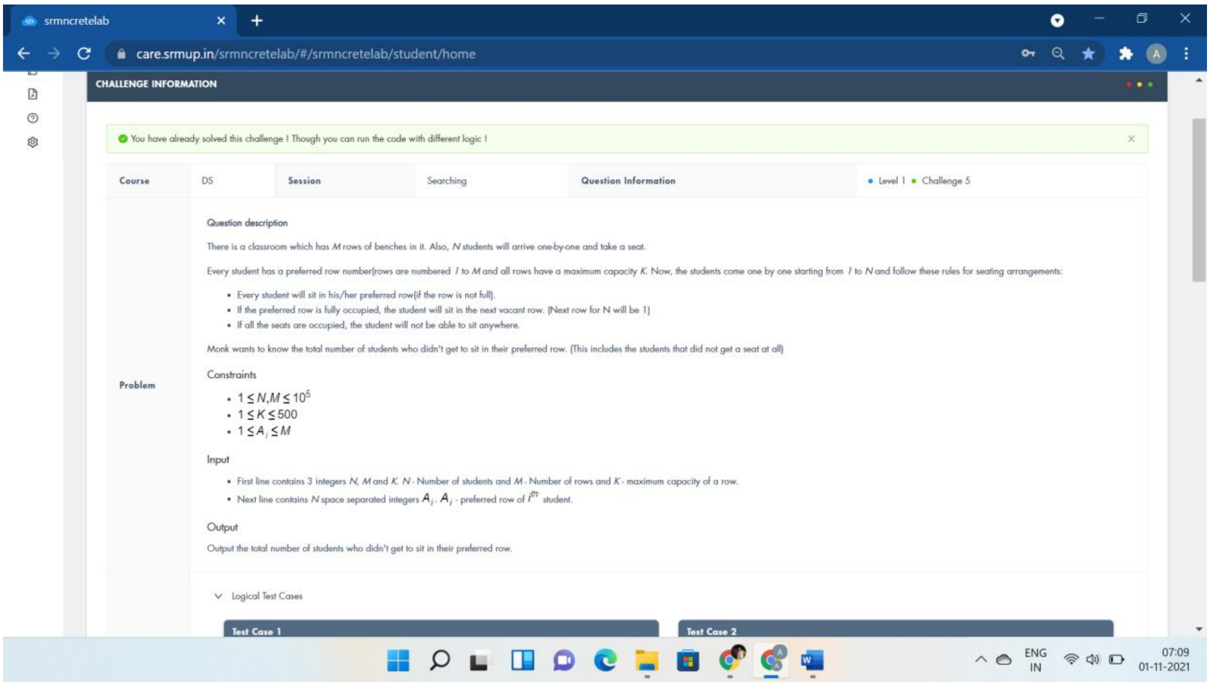
INPUT (STDIN)

```
256
```

EXPECTED OUTPUT

```
#include <stdio.h>
int check(int s){
        int n,sum = 0;
        for (n = 1; sum < s; n++) {
                sum += n;
                if (sum == s)
```

```
                        return 1;
        }
        return -1;
}
int binarySearch(int low,int high,int key)
{
  return 1;
}
int main() {
 int n, i, flag = 0;
 scanf("%d", &n);
 for (i = 2; i <= n / 2; ++i) {
  if (check(i) == 1) {
   if (check(n - i) == 1) {
    flag = 1;
   }
  }
 }
 binarySearch(1,1,1);
 if (flag == 0)
  printf("NO");
 else
 printf("YES");
 return 0;
}
```



CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	● Level 1 ● Challenge 5

Problem

Question description

There is a classroom which has M rows of benches in it. Also, N students will arrive one-by-one and take a seat.

Every student has a preferred row number(rows are numbered 1 to M and all rows have a maximum capacity K. Now, the students come one by one starting from 1 to N and follow these rules for seating arrangements:

- Every student will sit in his/her preferred row(if the row is not full).
- If the preferred row is fully occupied, the student will sit in the next vacant row. (Next row for N will be 1)
- If all the seats are occupied, the student will not be able to sit anywhere.

Monk wants to know the total number of students who didn't get to sit in their preferred row. (This includes the students that did not get a seat at all)

Constraints

- $1 \leq N,M \leq 10^5$
- $1 \leq K \leq 500$
- $1 \leq A_i \leq M$

Input

- First line contains 3 integers N, M and K. N - Number of students and M - Number of rows and K - maximum capacity of a row.
- Next line contains N space separated integers A_i. A_i - preferred row of i^{th} student.

Output

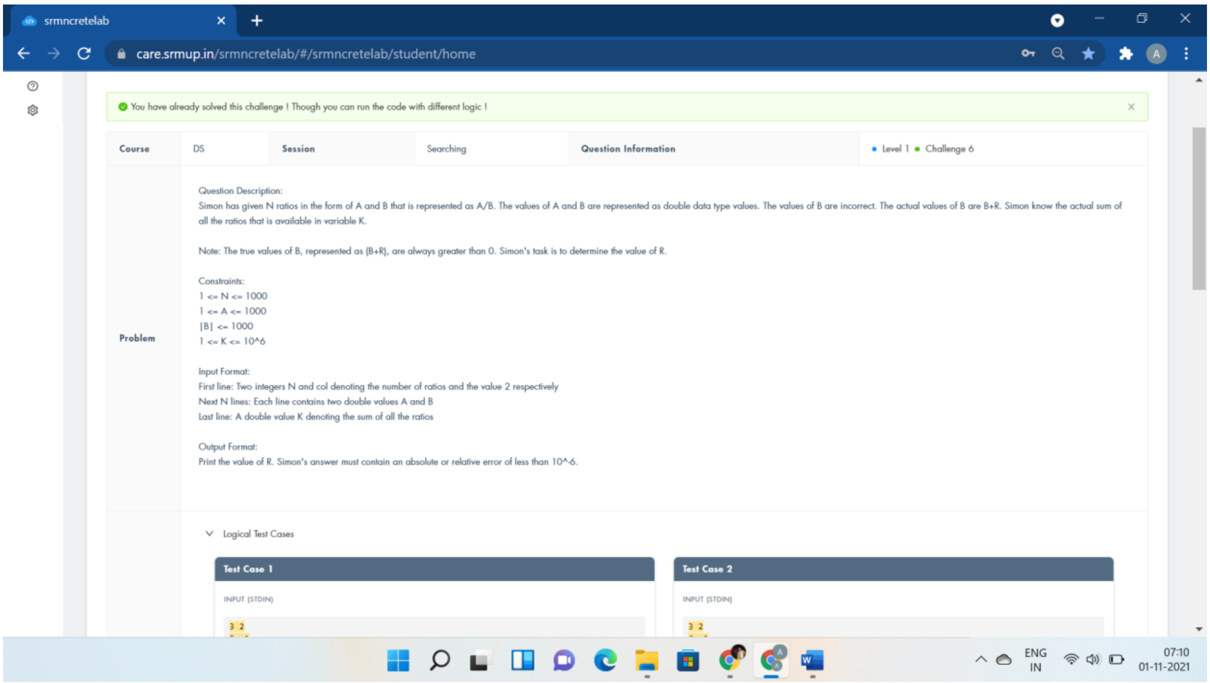
Output the total number of students who didn't get to sit in their preferred row.

Logical Test Cases

Test Case 1

Test Case 2

```
#include <stdio.h>


int main()

{

   int n,m,k,x,y,i,ans=0,flag=1;

   scanf("%d %d %d",&n,&m,&k);

   int a[100001]={0},b[100001]={0};

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

   {

      scanf("%d",&x);

      if(a[x]<k)

      {

         ans++;

         a[x]++;

      }

      else if(flag!=0)

      {

         y=x;

         x++;

         if(b[y]!=0)

         x=b[y];

         flag=0;

         while(x!=y)
      {
        if(x==m+1)
        x=1;
        if(x==y)
        break;
        if(a[x]<k)
        {
          a[x]++;
          flag=1;
          b[y]=x;
          break;
        }
        x++;
      }
    }
  }
  printf("%d",n-ans);
  return 0;
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1 Challenge 6

Problem

Question Description:
Simon has given N ratios in the form of A and B that is represented as A/B. The values of A and B are represented as double data type values. The values of B are incorrect. The actual values of B are B+R. Simon know the actual sum of all the ratios that is available in variable K.

Note: The true values of B, represented as (B+R), are always greater than 0. Simon's task is to determine the value of R.

Constraints:
1 <= N <= 1000
1 <= A <= 1000
|B| <= 1000
1 <= K <= 10^6

Input Format:
First line: Two integers N and col denoting the number of ratios and the value 2 respectively
Next N lines: Each line contains two double values A and B
Last line: A double value K denoting the sum of all the ratios

Output Format:
Print the value of R. Simon's answer must contain an absolute or relative error of less than 10^-6.

Logical Test Cases

Test Case 1

INPUT (STDIN)

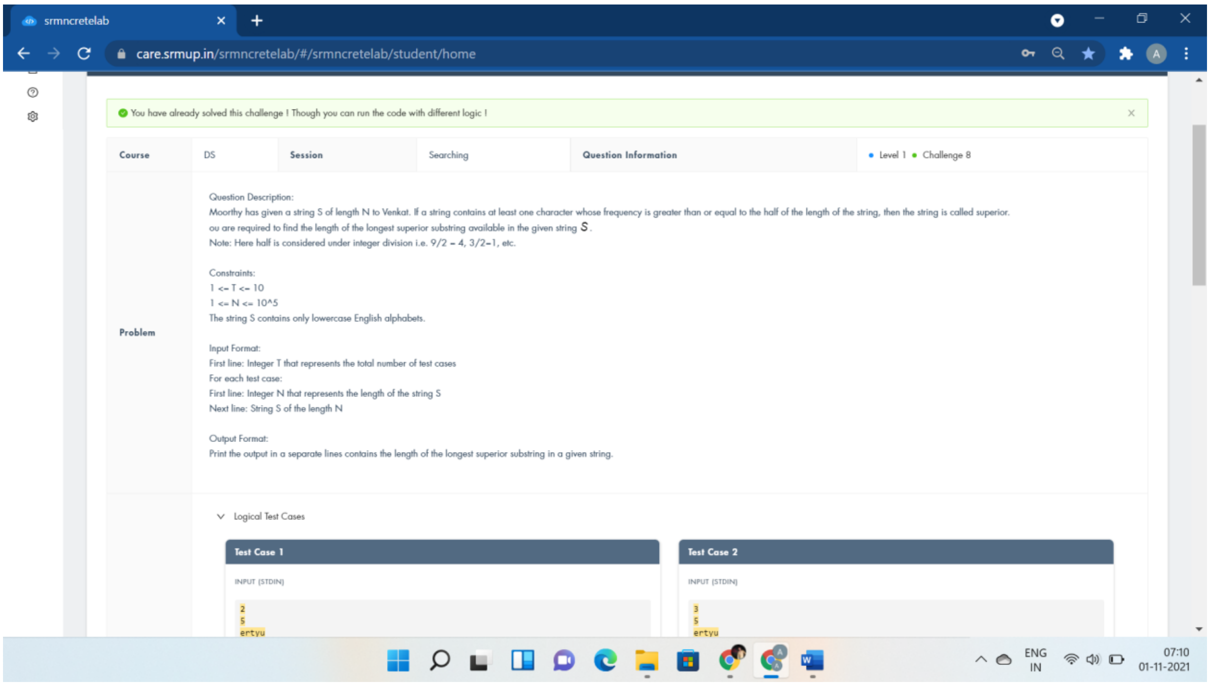
```
3 2
```

Test Case 2

INPUT (STDIN)

```
3 2
```

```
#include<iostream>

using namespace std;

double func(double arr[][2],double r,int n){
    double ans = 0;
    for (int i = 0; i < n; i++) {
        ans+= (arr[i][0]/(arr[i][1]+r));
    }
    return ans;
}
int main(){
    int n,two;
    cin>>n>>two;
    double arr[n][2];
    for (int i = 0; i < n; i++) {
        cin>>arr[i][0]>>arr[i][1];
    }
    double hi=2000,lo=0,mid,curr,k;
    cin>>k;
    while(hi-lo>1e-7){
        mid=(hi+lo)/2;
        curr=func(arr,mid,n);
        if(curr<k){
            hi = mid;
        }
        else{
            lo = mid + 1e-7;
        }
    }
    printf("%.6f",mid);

    return 0;
    printf("double solve(double** arr,double K,int n)");
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	• Level 1 • Challenge 8

Problem

Question Description:
Moorthy has given a string S of length N to Venkat. If a string contains at least one character whose frequency is greater than or equal to the half of the length of the string, then the string is called superior.
ou are required to find the length of the longest superior substring available in the given string S.
Note: Here half is considered under integer division i.e. 9/2 = 4, 3/2=1, etc.

Constraints:
1 <= T <= 10
1 <= N <= 10^5
The string S contains only lowercase English alphabets.

Input Format:
First line: Integer T that represents the total number of test cases
For each test case:
First line: Integer N that represents the length of the string S
Next line: String S of the length N

Output Format:
Print the output in a separate lines contains the length of the longest superior substring in a given string.

Logical Test Cases

Test Case 1

INPUT (STDIN)

```
2
5
ertyu
```

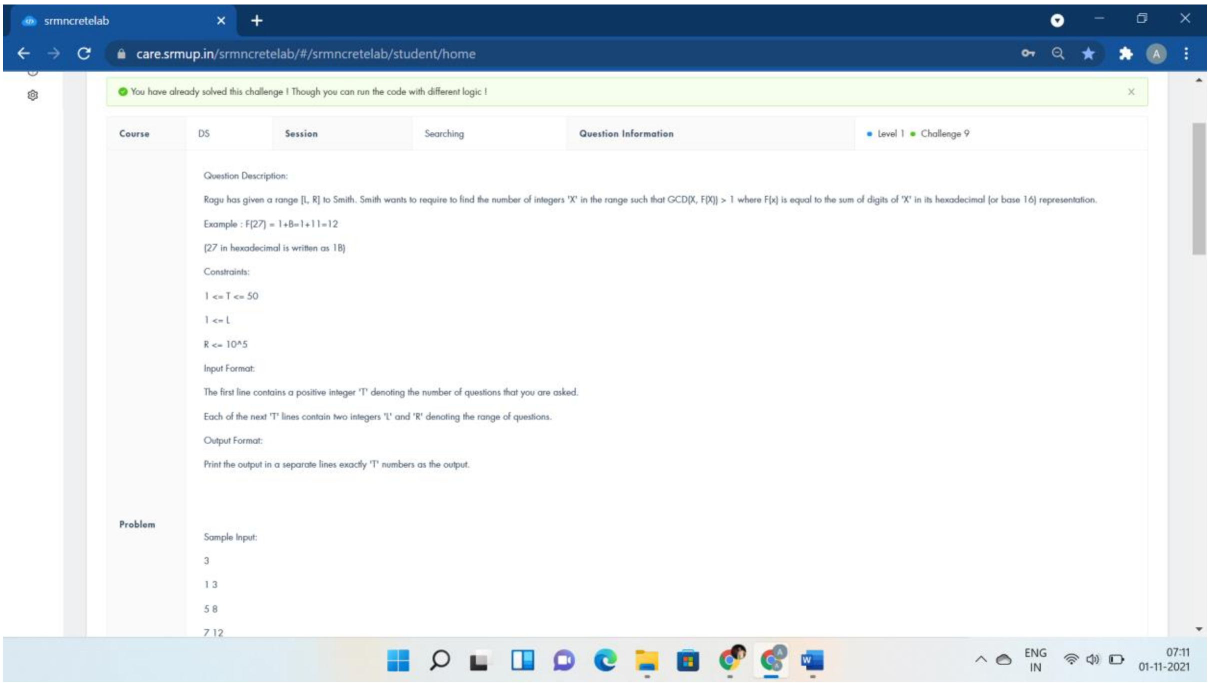
Test Case 2

INPUT (STDIN)

```
3
5
ertyu
```

```
#include <stdio.h>

void x()

{

   if(0)printf("int findmax(int* Count)");

}

int main()

{

   int t,i,j;

   scanf("%d",&t);

   while(t--)

   {

      int n;

      scanf("%d",&n);

      char s[n],c[26]={0};

      scanf("%s",s);

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

      {

         j=(int)s[i]-97;

         c[j]++;

      }

      j=0;

      for(i=0;i<26;i++)

      if(c[i]>j)
```

```
    j=c[i];

    printf("%d\n",j*2+1);

  }
        return 0;
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1 Challenge 9

Problem

Question Description:

Ragu has given a range [L, R] to Smith. Smith wants to require to find the number of integers 'X' in the range such that GCD(X, F(X)) > 1 where F(x) is equal to the sum of digits of 'X' in its hexadecimal (or base 16) representation.

Example : F(27) = 1+B=1+11=12

(27 in hexadecimal is written as 1B)

Constraints:

1 <= T <= 50

1 <= L

R <= 10^5

Input Format:

The first line contains a positive integer 'T' denoting the number of questions that you are asked.

Each of the next 'T' lines contain two integers 'L' and 'R' denoting the range of questions.

Output Format:

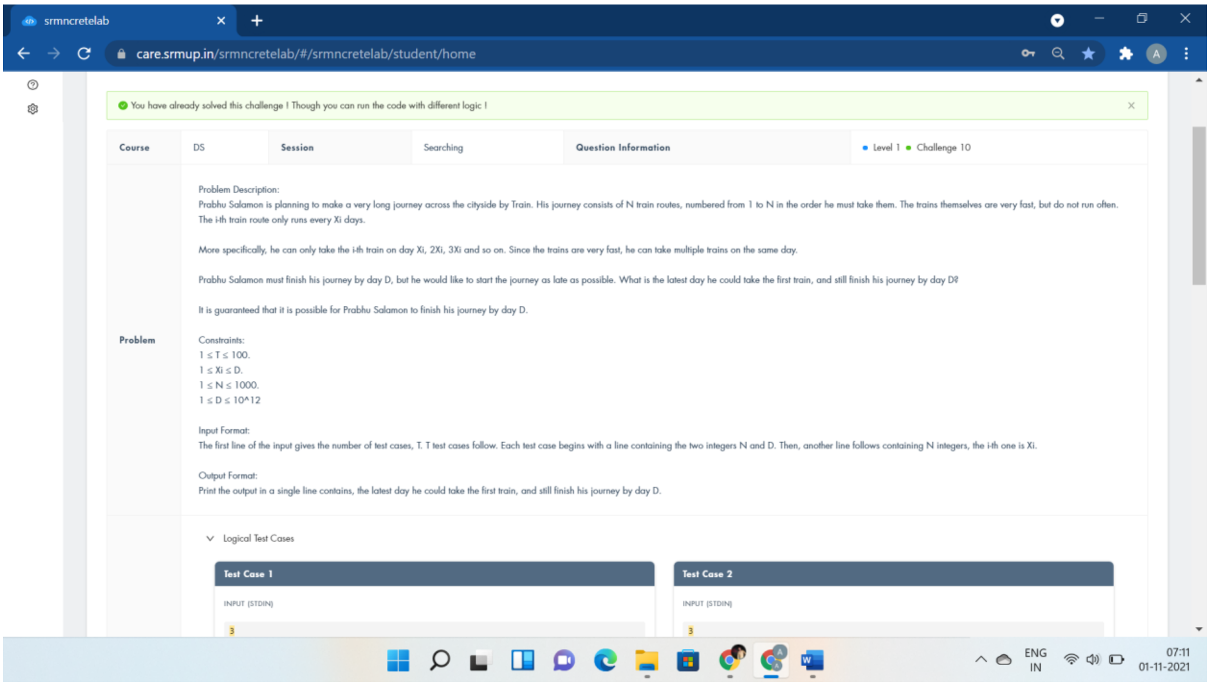
Print the output in a separate lines exactly 'T' numbers as the output.

Sample Input:

```
3
1 3
5 8
7 12
```

```
#include<bits/stdc++.h>
using namespace std;
int F(int x){
  int sum = 0;
        while(x > 0){
                sum += x%16;
                x = x/16;
        }
        return sum;
}
int search(int a, int b){
  int count=0;
    for(int i=a;i<=b;i++){
      if(__gcd(i,F(i))>1)
```

```
            count++;
        }
        return count;
}
int main(){
    int t,l,r;
    cin>>t;
    while(t--){
        cin>>l>>r;
        //int count=0;
        //for(int i=l;i<=r;i++){
        //    if(__gcd(i,F(i))>1)
        //    count++;
       // }
       int count=search(l,r);
        cout<<count<<endl;
    }
}
```



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 10

Problem

Problem Description:
Prabhu Salamon is planning to make a very long journey across the cityside by Train. His journey consists of N train routes, numbered from 1 to N in the order he must take them. The trains themselves are very fast, but do not run often. The i-th train route only runs every Xi days.

More specifically, he can only take the i-th train on day Xi, 2Xi, 3Xi and so on. Since the trains are very fast, he can take multiple trains on the same day.

Prabhu Salamon must finish his journey by day D, but he would like to start the journey as late as possible. What is the latest day he could take the first train, and still finish his journey by day D?

It is guaranteed that it is possible for Prabhu Salamon to finish his journey by day D.

Constraints:
1 ≤ T ≤ 100.
1 ≤ Xi ≤ D.
1 ≤ N ≤ 1000.
1 ≤ D ≤ 10^12

Input Format:
The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the two integers N and D. Then, another line follows containing N integers, the i-th one is Xi.

Output Format:
Print the output in a single line contains, the latest day he could take the first train, and still finish his journey by day D.

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main() {
    int T, n, d;
    cin >> T;
    for(int t=0;t<T;t++) {
        cin >> n >> d;
        stack <int> bus;
        for(int i=n-1;i>=0;i--){
            int x;
            cin >> x;
            bus.push(x);
        }
        while(!bus.empty()){
            int b = bus.top();
            bus.pop();
            d = d -  d%b;
        }
        cout<<d<< endl;
        }
        return 0;
}
```