

**Q. NEW or OLD**

When parents take their kids for Engineering Counselling, they always go with a preconceived notion that older the college, better will be the quality of education offered. There is a help desk in front of the counselling hall to tell out of the colleges in which seats are available, which college is the older one.

Nowadays, engineering counselling goes on for a month and the help desk needs to function on all days. So the Dean, Admissions decided to automate this task. Can you help him in this task?

Given the year of establishment of 2 colleges, write a program to determine which college is the older one.

**Input Format:**

Input consists of 2 integers. The first integer corresponds to the year of establishment of college 1 and the second integer corresponds to the year of establishment of college 2.

**Output Format:**

Output consists of the string College 1 is older or College 2 is older.

Refer sample input and output for further formatting specifications..

**Source Code**

```
#include <stdio.h>
int main()
{
    int y1,y2;
    scanf("%d%d",&y1,&y2);
    if(y1>y2)
        printf("College 2 is older");
    else
        printf("College 1 is older");
    return 0;
}
```

**Sample Input**

2002  
2008

**Sample Output**

College 1 is older

**Result**

Thus, Program " **NEW or OLD** " has been successfully executed

### **Q. Number Reversing**

Kamalas teacher give her mobile number in reverse order. She asks kamala to rearrange the number using c program. Now kamala needs your help to write a c code for reversing the number.

#### **Source Code**

```
#include <stdio.h>
int main(){
    int n,rev = 0, remainder;
    scanf("%d",&n);
    while (n != 0) {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n /= 10;
    }
    printf("%d",rev);
    return 0;
}
```

#### **Sample Input**

256

#### **Sample Output**

652

#### **Result**

Thus, Program " **Number Reversing** " has been successfully executed

**Q. IO 12**

Write a program to print the ASCII value of a character

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%c",&c);
    printf("ASCII value of %c=%d",c,c);
    return 0;
}
```

**Sample Input**

c

**Sample Output**

ASCII value of c=99

**Result**

Thus, Program " **IO 12** " has been successfully executed

#### **Q. Heena Challenge**

Heena gave a number to Jonnes and ask him to count the number of digits in a number. Sheela tries one simple logic to do that. So she decided to apply this concept in C.

#### **Source Code**

```
#include <stdio.h>
int main()
{
    int n,count=0;
    scanf("%d",&n);
    while(n>0)
    {
        n=n/10;
        count=count+1;
    }
    printf("%d",count);

    return 0;
}
```

#### **Sample Input**

8952

#### **Sample Output**

4

#### **Result**

Thus, Program " **Heena Challenge** " has been successfully executed

#### Q. Adding two distances

1. Create a Structure called "Distance"
2. Create two data members of "Distance Structure" feet(int), inch(float)
3. Create three structure variables as d1, d2 and sumOfDistances 4. Get two distances and add the feet and inches.

Mandatory:

To add the distance using the structure variables as follows

1. sumOfDistances.feet=d1.feet+d2.feet
- 2 sumOfDistances.inch=d1.inch+d2.inch

#### Source Code

```
#include <stdio.h>
struct Distance
{
    int feet;
    float inch;
}d1,d2,sumOfDistances;
int main()
{
    scanf("%d %f",&d1.feet,&d1.inch);
    scanf("%d %f",&d2.feet,&d2.inch);
    {
        sumOfDistances.feet=d1.feet+d2.feet;
        sumOfDistances.inch=d1.inch+d2.inch;
        printf("Sum of distances=%d feet and %.2f inches",sumOfDistances.feet,sumOfDistances.inch);
    }
    return 0;
}
```

#### Sample Input

```
23 8.6
34 2.4
```

#### Sample Output

```
Sum of distances=57 feet and 11.00 inches
```

#### Result

Thus, Program " **Adding two distances** " has been successfully executed

**Q. Structure - 1 Students Details**

Create a structure called Student.

```
struct Student
{
    char name[30];
    char department[20];
    int yearOfStudy;
    float cgpa;
};

Input and Output Format:
Peter sample input and output for formatting specification.
All text fields corresponds to input.
Name, Department, Year of study, CGPA.
The rest corresponds to output.
```

Students details are sorted based on their "Names" in ascending order

**Source Code**

```
#include <stdio.h>
#include <string.h>
struct Student
{
    char name[30];
    char department[20];
    int yearOfStudy;
    float cgpa;
};

int main()
{
    struct Student S[1000];
    struct Student t;
    int i,j,n;
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        scanf("%s", S[i].name);
        scanf("%s", &S[i].department);
        scanf("%d", &S[i].yearOfStudy);
        scanf("%f", &S[i].cgpa);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(strcmp(S[i].name,S[j].name)>0)
            {
                t=S[i];
                S[i]=S[j];
                S[j]=t;
            }
        }
    }

    for(i=0;i<n;i++)
    {
        printf("Name:%s\n", S[i].name);
        printf("Department:%s\n", S[i].department);
        printf("Year of study:%d\n", S[i].yearOfStudy);
        printf("CGPA:%.2f\n", S[i].cgpa);
    }
    return 0;
}
```

**Sample Input**

```
3
raju cse 1.78
somu IT 2.82
Jagan swe 3.8.6
```

**Sample Output**

```
Name:jagan
Department:swe
Year of study:3
CGPA:8.6
Name:raju
Department:cse
Year of study:1
CGPA:7.8
Name:somu
Department:IT
Year of study:2
CGPA:8.2
```

**Result**

Thus, Program " Structure - 1 Students Details " has been successfully executed

#### Q. How many bits?

Sasha ask Diya to write a program to get minimum number of bits to store an integer number. Devi thinks a lot but she could not able to do this. Can you help her to derive a solution for this?

#### Source Code

```
#include <stdio.h>
int countBit(int);
int main()
{
    int num;
    scanf("%d",&num);

    printf("%d\n",countBit(num));
    return 0;
}

int countBit(int n)
{
    int count=0,i;
    if(n==0) return 0;
    for(i=0; i< 32; i++)
    {
        if( (1 << i) & n)
            count=i;
    }
    return ++count;
}
```

#### Sample Input

127

#### Sample Output

7

#### Result

Thus, Program " How many bits? " has been successfully executed

#### **Q. Pointers - 48**

Write a program to reverse a string using pointer

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

#### **Source Code**

```
#include <stdio.h>
#include<string.h>
int main()
{
    char str[100],revStr[100];
    int i,j;
    scanf("%[^\\n]s",str);
    j=0;
    for(i=(strlen(str)-1);i>=0;i--)
        revStr[j++]=str[i];
    revStr[j]='\0';
    printf("%s",revStr);
    return 0;
}
```

#### **Sample Input**

SRM university

#### **Sample Output**

ytsrevinu MRS

#### **Result**

Thus, Program " **Pointers - 48** " has been successfully executed

### Q. Caravan

Most problems on CodeChef highlight chefs love for food and cooking but little is known about his love for racing sports. He is an avid Formula 1 fan. He went to watch this years Indian Grand Prix at New Delhi. He noticed that one segment of the circuit was a long straight road. It was impossible for a car to overtake other cars on this segment. Therefore, a car had to lower down its speed if there was a slower car in front of it. While watching the race, Chef started to wonder how many cars were moving at their maximum speed.

Formally, you're given the maximum speed of N cars in the order they entered the long straight segment of the circuit. Each car prefers to move at its maximum speed. If that's not possible because of the front car being slow, it might have to lower its speed. It still moves at the fastest possible speed while avoiding any collisions. For the purpose of this problem, you can assume that the straight segment is infinitely long.

Count the number of cars which were moving at their maximum speed on the straight segment.

Input

The first line of the input contains a single integer T denoting the number of test cases to follow. Description of each test case contains 2 lines. The first of these lines contain a single integer N, the number of cars. The second line contains N space separated integers, denoting the maximum speed of the cars in the order they entered the long straight segment.

Output

For each test case, output a single line containing the number of cars which were moving at their maximum speed on the segment.

### Source Code

```
#include<stdio.h>
#include<stdint.h>
int main(){
    int t,i;
    scanf("%d",&t);
    while (t--) {
        long int n;
        scanf("%d",&n);
        int32_t a[n];
        int32_t c;
        long int count=0;
        for (i = 0; i < n; i++)
            scanf("%d",&a[i]);
        c = a[0];
        for (i = 0; i < n; i++) {
            if (a[i] <= c) {
                c = a[i];
                count++;
            }
        }
        printf("%d\n",count);
    }
    return 0;
}
```

### Sample Input

```
3
1
10
3
8 3 6
5
4 5 1 2 3
```

### Sample Output

```
1
2
2
```

### Result

Thus, Program " Caravan " has been successfully executed

#### Q. Adding two distances

1. Create a Structure called "Distance"
2. Create two data members of "Distance Structure" feet(int), inch(float)
3. Create three structure variables as d1, d2 and sumOfDistances
4. Get two distances and add the feet and inches.

Mandatory:

To add the distance using the structure variables as follows

1. sumOfDistances.feet=d1.feet+d2.feet
- 2 sumOfDistances.inch=d1.inch+d2.inch

#### Source Code

```
#include <stdio.h>
struct Distance
{
    int feet;
    float inch;
}d1,d2,sumOfDistances;
int main()
{
    scanf("%d %f\n",&d1.feet,&d1.inch);
    scanf("%d %f\n",&d2.feet,&d2.inch);
    {
        sumOfDistances.feet=d1.feet+d2.feet;
        sumOfDistances.inch=d1.inch+d2.inch;
        printf("Sum of distances=%d feet and %.2f inches",sumOfDistances.feet,sumOfDistances.inch);
    }
    return 0;
}
```

#### Sample Input

23 8.6  
34 2.4

#### Sample Output

Sum of distances=57 feet and 11.00 inches

#### Result

Thus, Program " **Adding two distances** " has been successfully executed

#### Q. Even or odd

Somesh and sakthi played one game to find the number is even or not, for that they have designed one coding, using union concept track yourself by your code to challenge them for their inputs.

Input Method

Integer ranges from 1 to 999

Output Method

Print the number is even or not

#### Source Code

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%2==0)
        printf("Even");
    else
        printf("Odd");
    return 0;
}
```

#### Sample Input

2

#### Sample Output

Even

#### Result

Thus, Program " **Even or odd** " has been successfully executed

**Q. Berland University**

There are N students living in the dormitory of Berland State University. Each of them sometimes wants to use the kitchen, so the head of the dormitory came up with a timetable for kitchens usage in order to avoid the conflicts:

The first student starts to use the kitchen at the time 0 and should finish the cooking not later than at the time A1.

The second student starts to use the kitchen at the time A1 and should finish the cooking not later than at the time A2.

And so on... The N-th student starts to use the kitchen at the time AN-1 and should finish the cooking not later than at the time AN.

The holidays in Berland are approaching, so today each of these N students wants to cook some pancakes. The i-th student needs Bi units of time to cook.

Input

The students have understood that probably not all of them will be able to cook everything they want. How many students will be able to cook without violating the schedule?

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.

The first line of each test case contains a single integer N denoting the number of students.

The second line contains N space-separated integers A1, A2, ..., AN denoting the moments of time by when the corresponding student should finish cooking.

Output

The third line contains N space-separated integers B1, B2, ..., BN denoting the time required for each of the students to cook.

For each test case, output a single line containing the number of students that will be able to finish the cooking.

Constraints

Should contain all the constraints on the input data that you may have. Format it like:

```
1 <= N <= 10
0 <= Ai <= 104
0 <= Bi <= 109
0 <= AN < 109
```

**Source Code**

```
#include <stdio.h>
int main()
{
    int T,N;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&N);
        int A[N+1];
        for(i=0;i<N;i++){
            scanf("%d",&A[i]);
        }
        int B[N+1];
        for(i=0;i<N;i++){
            scanf("%d",&B[i]);
        }
        int count=0,temp=A[0];
        for(i=0;i<N;i++){
            if(B[i]<=temp){
                ++count;
            }
            temp=A[i+1];
        }
        printf("%d\n",count);
    }
    return 0;
}
```

**Sample Input**

```
2
3
1 10 15
1 10 3
3
10 20 30
15 5 20
```

**Sample Output**

```
2
1
```

**Result**

Thus, Program " Berland University " has been successfully executed

### Q. IO 32

Manav had a play with his friend gourav, he gave a number that to read a floating point number and asked him to display the right most digit integral part of the number. Please help gourav to display the right most integral part of the number.

Explanation :

If the input is given 124.34, then the output to be displayed is 4 (i.e) Before decimal the integral part is 124 , in that last digit is 4.

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

### Source Code

```
#include <stdio.h>
int main()
{
    float num;
    int x;
    scanf("%f",&num);
    x=(int)num;
    printf("Rightmost integer digit of %.2f=%d",num,x%10);
    return 0;
}
```

### Sample Input

233.444

### Sample Output

Rightmost integer digit of 233.44=3

### Result

Thus, Program " **IO 32** " has been successfully executed

**Q. BIGGEST VALUE IN A GIVEN ARRAY**

Find biggest value in the array using pointers in C

**Source Code**

```
#include <stdio.h>
int main() {
    int a[100],n,c,maximum;
    scanf("%d",&n);
    for(c=0;c<n;c++)
        scanf("%d",&a[c]);
    maximum=a[0];
    for(c=1;c<n;c++)
    {
        if(a[c]>maximum)
        {
            maximum=a[c];
        }
    }
    printf("%d",maximum);
    return 0;
    return 0;
}
```

**Sample Input**

2  
58 98

**Sample Output**

98

**Result**

Thus, Program " **BIGGEST VALUE IN A GIVEN ARRAY** " has been successfully executed

**Q. REVERSE STRING**

Reverse String using Pointers in C

**Source Code**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100],rev[50];
    int i,len;
    scanf("%s",str);
    char *sptr=str;
    char *rptr=rev;
    len=strlen(str);
    for(i=len-1;i>=0;i--)
    {
        printf("%c",str[i]);
    }
    return 0;
}
```

**Sample Input**

Cse

**Sample Output**

esC

**Result**

Thus, Program " **REVERSE STRING** " has been successfully executed

**Q. SUM OF 6 NUMBERS**

Print the sum of 6 numbers using array and pointers

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,sum=0,i;
    for(i=0;i<6;i++)
    {
        scanf("%d",&a);
        sum=sum+a;
    }
    printf("%d",sum);
    return 0;
}
```

**Sample Input**

1 2 3 4 5 6

**Sample Output**

21

**Result**

Thus, Program " **SUM OF 6 NUMBERS** " has been successfully executed

**Q. Pointer - 26**

Write a program to compare two arrays using pointers.

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,i,j,f;
    scanf("%d",&a);
    int arr[a];
    for(j=0;j<a;j++)
    {
        scanf("%d",&arr[j]);
    }
    scanf("%d",&b);
    int arr1[b];
    for(i=0;i<b;i++)
    {
        scanf("%d",&arr1[i]);
    }
    for(i=0;i<a;i++)
    {
        if(arr[i]==arr1[i])
        {
            f=1;
            break;
        }
    }
    if(f==1)
        printf("Arrays are not equal");
    else
        printf("Arrays are equal");

    return 0;
}
```

**Sample Input**

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

**Sample Output**

Arrays are not equal

**Result**

Thus, Program " **Pointer - 26** " has been successfully executed

**Q. Pointers - 24**

Write a function that accepts a string using pointers. In the function ,delete all the occurrences of a given character and display the modified string on the screen

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[15],ch,cat[10];
    scanf("%s%s",str,cat);
    scanf("%s",&ch);
    int i=0,j,len;
    len=strlen(str);
    for(i=0;i<len;i++)
    {
        if(str[i]==ch)
        {
            for(j=i;j<len;j++)
            {
                str[j]=str[j+1];
            }
            len--;
            i--;
        }
    }
    printf("%s ",str);
    printf("%s",cat);
    return 0;
}
```

**Sample Input**

SRM University  
S

**Sample Output**

RM University

**Result**

Thus, Program " **Pointers - 24** " has been successfully executed

**Q. SWAPPING**

Your task is to write a code to swapping of two numbers

|

Input:

Input should contain the value of the limit n

|

Output:

It should print the Sum of series upto n limit

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,t;
    scanf("%d%d",&a,&b);
    t=a;
    a=b;
    b=t;
    printf("%d %d",a,b);
    return 0;
}
```

**Sample Input**

12 89

**Sample Output**

89 12

**Result**

Thus, Program " SWAPPING " has been successfully executed

**Q. Pointers - 30**

Write a program which takes an input from the user and then checks whether its a number or a character . If its a character ,determine whether it is in upper case or lower case

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%s",&c);
    if(c>='A' && c<='Z')
    {
        char z=c+32;
        printf("Input is upper case\n");
        printf("Lower case=%s",&z);
    }
    else if(c>='a' && c<='z')
    {
        char x=c-32;
        printf("Input is lower case\n");
        printf("Upper case=%s",&x);
    }
    return 0;
}
```

**Sample Input**

S

**Sample Output**

Input is upper case  
Lower case=s

**Result**

Thus, Program " **Pointers - 30** " has been successfully executed

**Q. Pointer - 2**

Write a program to enter a lowercase character. Print this character in uppercase and also display its ASCII value

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%c",&c);
    printf("The ASCII value is:%d\n",c);
    printf("The uppercase letter is:%c",c-32);
    return 0;
}
```

**Sample Input**

c

**Sample Output**

The ASCII value is:99

The uppercase letter is:C

**Result**

Thus, Program " **Pointer - 2** " has been successfully executed

### Q. Sum of cube of Digits

Ganesh played game with his friend sanjay to find Sum of Cube of digits of the number he said. Help to Ganesh and sanjay to solve it by your code using union.

Input Method

Integer ranges from 1 to 999

Output Method

Sum of Cube of digits of the number

### Source Code

```
#include <stdio.h>
struct cub
{
    int a;
}c;
int main()
{
    int sum=0,digit=0;
    scanf("%d",&c.a);
    while(c.a>0)
    {
        digit=c.a%10;
        sum=sum+digit;
        c.a=c.a/10;
    }
    printf("Sum=%d\n",sum);
    printf("Cube=%d",sum*sum*sum);
    return 0;
}
```

### Sample Input

123

### Sample Output

Sum=6  
Cube=216

### Result

Thus, Program " Sum of cube of Digits " has been successfully executed

**Q. Even or odd**

Somesh and sakthi played one game to find the number is even or not, for that they have designed one coding, using union concept track yourself by your code to challenge them for their inputs.

Input Method

Integer ranges from 1 to 999

Output Method

Print the number is even or not

Mandatory:

Use Structure Concept

**Source Code**

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%2==0)
        printf("Even");
    else
        printf("Odd");
    return 0;
}
```

**Sample Input**

2

**Sample Output**

Even

**Result**

Thus, Program " **Even or odd** " has been successfully executed

**Q. Find no of characters in name**

Manoj arranged one event to find no of characters in his friends name, your idea is to give your friends name, for that manoj has to answer the no of characters present in it, with the help of sturcture concept accomplish it.

Input Method

Name of different friends

Output Method

No of characters

Mandatory:

Use Structure Concepts

**Source Code**

```
#include <stdio.h>
#include <string.h>
struct name
{
    char str[10];
} s;
int main()
{
    int len;
    char str[10];
    scanf("%s",str);
    struct name;
    len=strlen(str);
    printf("%d",len);
    return 0;
}
```

**Sample Input**

raja

**Sample Output**

4

**Result**

Thus, Program " **Find no of characters in name** " has been successfully executed

**Q. Payroll using Structures**

1. Create a Structure "employee"
2. Create six data members for structures as name(char), empid(int), salary(int), hra(int), da(int), total(float)
3. Input the data of the employee as name, empid, salary.
4. Calculate the HRA(10% salary), DA(20% salary)
5. Total pay = salary +hra +da
6. Create structure variable as "emp"

**Source Code**

```
#include <stdio.h>
struct employee
{
    int empid,salary,hra,da;
    char name[20];
    float total;
}emp;
int main()
{
    scanf("%s",emp.name);
    scanf("%d",&emp.empid);
    scanf("%d",&emp.salary);
    emp.hra=emp.salary*0.1;
    emp.da=emp.salary*0.2;
    printf("Name=%s",emp.name);
    printf("\nId=%d",emp.empid);
    printf("\nHRA=%d",emp.hra);
    printf("\nDA=%d",emp.da);
    printf("\nTotal Salary=%f",emp.salary+(emp.salary*0.1)+(emp.salary*.2));
    return 0;
}
```

**Sample Input**

Bogar  
1000  
15000

**Sample Output**

Name=Bogar  
Id=1000  
HRA=1500  
DA=3000  
Total Salary=19500

**Result**

Thus, Program " **Payroll using Structures** " has been successfully executed

**Q. Nested Structure - Details of Student**

Create two structures as follows:

Structure 1:

Name = student  
Data members = name(char), rollno(int)

Structure 2:

Name = dateOfBirth  
Data members = date(int), month(int), year(int)  
Create Structure Variable as "DOB"

In main function:

1. Create structure variable for student "std"  
Hint: struct student std;

2. Input the values of name and roll number  
Hint: std.name

3. Input the values of date, month and year  
Hint: std.DOB.date

4. Display the name, roll number, Date of Birth.

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

**Source Code**

```
#include<stdio.h>
struct student
{
    char name[100];
    struct dateOfBirth
    {
        int rollno,date,month,year;
    }DOB;
}std;
int main()
{
    scanf("%s%d%d%d",std.name,&std.DOB.rollno,&std.DOB.date,&std.DOB.month,&std.DOB.year);
    printf("Name=%s\nRollNo=%d\nDate of birth=%d/%d/%d",std.name, std.DOB.rollno, std.DOB.date, std.DOB.month, std.DOB.year);
    return 0;
}
```

**Sample Input**

Rajesh  
101  
25 12 1989

**Sample Output**

Name=Rajesh  
RollNo=101  
Date of birth=25/12/1989

**Result**

Thus, Program " Nested Structure - Details of Student " has been successfully executed

**Q. Find no of characters in name**

Manoj arranged one event to find no of characters in his friends name, your idea is to give your friends name, for that manoj has to answer the no of characters present in it, with the help of sturcuture concept accomplish it.  
Input Method

Name of different friends

Output Method

No of characters

**Source Code**

```
#include <stdio.h>
#include <string.h>
struct name
{
    char str[10];
}
s;
int main()
{
    int len;
    char str[10];
    scanf("%s",str);
    struct name;
    len=strlen(str);
    printf("%d",len);
    return 0;
}
```

**Sample Input**

raja

**Sample Output**

4

**Result**

Thus, Program " **Find no of characters in name** " has been successfully executed

**Q. Student Management Systems**

1. In this program, create a structure, "student"
2. The structure has three members: name (string), roll (integer) and marks (float).

Then, a structure variable "s" is created to store information and display it on the screen.

3. Input the student information and display the details.

Mandatory:

1. Create a structure "student" and the structure variable as "s"
2. Print the information using structurevariable.members

Example (s.name)

**Source Code**

```
#include <stdio.h>
struct student
{
    char name[50];
    int roll;
    float marks;
}
s;
int main()
{
    scanf("%s",s.name);
    scanf("%d",&s.roll);
    scanf("%f",&s.marks);
    printf("\nName=%s",s.name);
    printf("\nRoll number=%d",s.roll);
    printf("\nMarks=%.2f",s.marks);
    return 0;
}
```

**Sample Input**

Bogar  
2000  
99.51

**Sample Output**

Name=Bogar  
Roll number=2000  
Marks=99.51

**Result**

Thus, Program " **Student Management Systems** " has been successfully executed

### Q. Time Machine Difference

The Time Machine is a science fiction novella by H. G. Wells, published in 1895 and written as a frame narrative. The work is generally credited with the popularization of the concept of time travel by using a vehicle that allows an operator to travel purposely and selectively forwards or backwards in time. The term "time machine", coined by Wells, is now almost universally used to refer to such a vehicle.

The Time Machine has been adapted into three feature films of the same name, as well as two television versions, and a large number of comic book adaptations. It has also indirectly inspired many more works of fiction in many media productions. Now kindly help our Captain of the Ship "H.G.Wells" to calculate the difference between the two time machines.

Mandatory:

1. Create a Structure "Time" and three data members as seconds(int), minutes(int), hours(int)
2. The Structure Variables as "startTime, stopTime, diff"
3. Access the data members as follows

stopTime.seconds  
startTime.hours

Note: The structure variables, data members and structure name are CASE Sensitive.

Follow the same case mentioned in the mandatory

### Source Code

```
#include <stdio.h>
struct time
{
    int seconds,minutes,hours;
};
void differenceBetweenTimePeriod(struct time t1,struct time t2,struct time*diff);
int main()
{
    struct time startTime,stopTime,diff;
    scanf("%d %d %d",&startTime.hours,&startTime.minutes,&startTime.seconds);
    scanf("%d %d %d",&stopTime.hours,&stopTime.minutes,&stopTime.seconds);
    differenceBetweenTimePeriod(startTime,stopTime,&diff);
    printf("TIME DIFFERENCE=%d:%d:%d",startTime.hours,startTime.minutes,startTime.seconds);
    printf("%d:%d:%d",stopTime.hours,stopTime.minutes,stopTime.seconds);
    printf("%d:%d:%d",diff.hours,diff.minutes,diff.seconds);
    return 0;
}
void differenceBetweenTimePeriod(struct time start,struct time stop,struct time *diff)
{
    if(stop.seconds>start.seconds)
    {
        --start.minutes;
        start.seconds+=60;
    }
    diff->seconds=stop.seconds-start.seconds;
    if(stop.minutes>start.minutes)
    {
        --start.hours;
        start.minutes+=60;
    }
    diff->minutes=stop.minutes-start.minutes;
    diff->hours=stop.hours-start.hours;
}
```

### Sample Input

```
12 34 55
8 12 15
```

### Sample Output

```
TIME DIFFERENCE=12:34:55-8:12:15=4:22:40
```

### Result

Thus, Program " **Time Machine Difference** " has been successfully executed

**Q. DIAGONAL MATRIX**

Diagonal Matrix.A square matrix which has zeros everywhere other than the main diagonal. Entries on the main diagonal may be any number, including 0.  
Write a program to find whether a given matrix is a diagonal matrix or not.

Input Format:

The input consists of  $(n \times n + 1)$  integers. The first integer corresponds to the number of rows/columns in the matrix. The remaining integers correspond to the elements in the matrix. The elements are read in row wise order, first row first, then second row and so on. Assume that the maximum value of m and n is 5.

Output Format:

Print yes if it is a diagonal matrix. Print no if it is not a diagonal matrix.

**Source Code**

```
#include <stdio.h>
int main()
{
    int x[10][10],nr,r,c,flag;
    scanf("%d",&nr);
    if(nr==nr)
    {
        for(r=0 ; r<nr ; r++)
            for(c=0 ; c<nr ; c++)
                scanf("%d" , &x[r][c]);

        flag=1 ;
        for(r=0 ; r<nr ; r++)
            for(c=0 ; c<nr ; c++)
                if(r==c)
                {
                    if(x[r][c]==0)
                        flag=0;
                }
                else
                {
                    if(x[r][c]!=0)
                        flag=0;
                }

        if(flag==1)
            printf("yes");
        else
            printf("no");
    }
    return 0;
}
```

**Sample Input**

```
3
4 5 3
5 4 1
0 0 3
```

**Sample Output**

```
no
```

**Result**

Thus, Program " **DIAGONAL MATRIX** " has been successfully executed

**Q. POWER OF 2**

Ram challenges Ganga to write a code to check whether an Integer Number is power of two (2) or not. Can you help Ganga to come out of this problem.

**Source Code**

```
#include <stdio.h>
int main()
{
    int num;
    int tempNum,flag;
    scanf("%d",&num);
    tempNum=num;
    flag=0;
    /*check power of two*/
    while(tempNum!=1)
    {
        if(tempNum%2!=0){
            flag=1;
            break;
        }
        tempNum=tempNum/2;
    }
    if(flag==0)
        printf("YES");
    else
        printf("NO");
    return 0;
}
```

**Sample Input**

32

**Sample Output**

YES

**Result**

Thus, Program " **POWER OF 2** " has been successfully executed

**Q. fun**

Add two numbers using user-defined function addNumbers() .

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("\n%d",a+b);
    return 0;
}
```

**Sample Input**

```
2
3
```

**Sample Output**

```
5
```

**Result**

Thus, Program " fun " has been successfully executed

**Q. Recursion 6 : GCD and LCM of 2 Numbers**

Write a program to compute GCD and LCM of 2 numbers using recursion.

Input and Output Format:

Input consists of 2 integers.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,x,y,temp,GCD,LCM;
    scanf("%d",&x);
    scanf("%d",&y);
    a=x;
    b=y;
    while(b!=0)
    {
        temp=b;
        b=a%b;
        a=temp;
    }
    GCD=a;
    LCM=(x*y)/GCD;
    printf("GCD=%d\n",GCD);
    printf("LCM=%d\n",LCM);

    return 0;
}
```

**Sample Input**

```
14
16
```

**Sample Output**

```
GCD=2
LCM=112
```

**Result**

Thus, Program "**Recursion 6 : GCD and LCM of 2 Numbers**" has been successfully executed

**Q. Sum of 2 arrays using Functions**

Write a program to find the sum of the corresponding elements in 2 arrays.

Input Format:

Input consists of  $2n+1$  integers. The first integer corresponds to  $n$ , the size of the array. The next  $n$  integers correspond to the elements in the first array. The last  $n$  integers correspond to the elements in the second array. Assume that the maximum value of  $n$  is 15.

Output Format:

Refer sample output for details

**Source Code**

```
#include <stdio.h>
int main()
{
    int n,a[100],b[100],i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
        scanf("%d",&b[i]);
    for(i=0;i<n;i++)
        printf("%d ",a[i]+b[i]);

    return 0;
}
```

**Sample Input**

```
5
2 3 6 8 2
1 1 1 1 1
```

**Sample Output**

```
3 4 7 9 3
```

**Result**

Thus, Program " **Sum of 2 arrays using Functions** " has been successfully executed

**Q. FUNSUMN**

Calculate sum of numbers 1 to N using recursion.

**Source Code**

```
#include <stdio.h>
int addnum(int n);
int main()
{
    int num;
    scanf("%d",&num);
    printf("%d",addnum(num));
    return 0;
}
int addnum(int n)
{
    if(n!=0)
        return n+addnum(n-1);
    else
        return n;
}
```

**Sample Input**

5

**Sample Output**

15

**Result**

Thus, Program " **FUNSUMN** " has been successfully executed

**Q. Rupee Name**

Write a program that reads the cost of an item in the form RRRR.PP (where RRRR denotes rupees and PP denotes paise)  
Refer sample input and output.

**Source Code**

```
#include <stdio.h>
int main()
{
    float Cost,Pai,Re;
    int Rup,i,R;
    scanf("%f",&Cost);
    Rup=Cost;
    Pai=(Cost-Rup)*100;
    i=Rup/100;
    {
        case 1:printf("ONE HUNDRED ");break;
        case 2:printf("TWO HUNDRED ");break;
        case 3:printf("THREE HUNDRED ");break;
        case 4:printf("FOUR HUNDRED ");break;
        case 5:printf("FIVE HUNDRED ");break;
        case 6:printf("SIX HUNDRED ");break;
        case 7:printf("SEVEN HUNDRED ");break;
        case 8:printf("EIGHT HUNDRED ");break;
        case 9:printf("NINE HUNDRED ");break;
    }
    i=Rup%100;
    R=i/10;
    switch(R)
    {
        case 1:printf("TEN");break;
        case 2:printf("TWENTY");break;
        case 3:printf("THIRTY");break;
        case 4:printf("FOURTY");break;
        case 5:printf("FIFTY");break;
        case 6:printf("SIXTY");break;
        case 7:printf("SEVENTY");break;
        case 8:printf("EIGHTY");break;
        case 9:printf("NINETY");break;
    }
    R=(Re-R)*10;
    switch(R)
    {
        case 1:printf("ONE");break;
        case 2:printf("TWO");break;
        case 3:printf("THREE");break;
        case 4:printf("FOUR");break;
        case 5:printf("FIVE");break;
        case 6:printf("SIX");break;
        case 7:printf("SEVEN");break;
        case 8:printf("EIGHT");break;
        case 9:printf("NINE");break;
    }
    printf(" AND PAISE ");
    i=Pai/10;
    Re=(Re-i)*10;
    switch(i)
    {
        case 1:printf("TEN");break;
        case 2:printf("TWENTY");break;
        case 3:printf("THIRTY");break;
        case 4:printf("FOURTY");break;
        case 5:printf("FIFTY");break;
        case 6:printf("SIXTY");break;
        case 7:printf("SEVENTY");break;
        case 8:printf("EIGHTY");break;
        case 9:printf("NINETY");break;
    }
    switch(R)
    {
        case 1:printf("ONE");break;
        case 2:printf("TWO");break;
        case 3:printf("THREE");break;
        case 4:printf("FOUR");break;
        case 5:printf("FIVE");break;
        case 6:printf("SIX");break;
        case 7:printf("SEVEN");break;
        case 8:printf("EIGHT");break;
        case 9:printf("NINE");break;
    }
    return 0;
}
```

**Sample Input**

104.3

**Sample Output**

ONE HUNDRED FOUR AND PAISE THIRTY

**Result**

Thus, Program " Rupee Name " has been successfully executed

### Q. Counting Vowels

Bruno likes to find out the vowels present in the string. He wants to count the number of vowels in the string. Help Bruno by writing a code.

### Source Code

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[100];
    int len,i,vow=0;
    scanf("%s",a);
    len=strlen(a);
    for(i=0;i<len;i++)
    {
        if(a[i]=='a'||a[i]=='A'||a[i]=='e'||a[i]=='E'||a[i]=='i'||a[i]=='I'||a[i]=='o'||a[i]=='O'||a[i]=='u'||a[i]=='U')
            vow=vow+1;
    }
    printf("%d",vow);
    return 0;
}
```

### Sample Input

success

### Sample Output

2

### Result

Thus, Program " **Counting Vowels** " has been successfully executed

**Q. Remove Repeated Characters in a String**

Write a C program to remove all repeated characters in a string

**Source Code**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];
    int i,j,k;
    scanf("%s",str);
    printf("%s\n",str);
    for(i = 0;i<strlen(str);i++)
    {
        for(j = i + 1;str[j]!='\0';j++)
        {
            if(str[j] == str[i])
            {
                for(k = j;str[k]!='\0';k++)
                {
                    str[k]=str[k + 1];
                }
            }
        }
    }
    printf("%s",str);
}

return 0;
}
```

**Sample Input**

srmuniversity

**Sample Output**

srmuniversity  
srmunivety

**Result**

Thus, Program " Remove Repeated Characters in a String " has been successfully executed

**Q. Substring**

You are given a string S of length N consisting only of 0s and 1s. You are also given an integer K. You have to answer Q queries. In the  $i$ th query, two integers  $L_i$  and  $R_i$  are given. Then you should print the number of substrings of  $S$  from index  $L_i$  to  $R_i$  such that it contains at least one 1. In other words, you have to count number of pairs  $[L, R]$  of integers such that  $L \leq R$  and substring  $S[L, R]$  occurs more than  $K$  times.

**Source Code**

```
#include <stdio.h>
#define _CRT_SECURE_NO_WARNINGS
char str[10005];
void getstr();
{
    int i=0;
    char ch='c';
    while(ch<'0'||ch>'9')
    {
        ch=getchar();
        if(i<<1)<(c<3)+ch>-0;
        ch=c;
        i++;
    }
    return ;
}

int main()
{
    int n,k,a,l,r,m,x,y,zeroone;
    long long int ans,a,b;
    int rat100005;
    int abc100005;
    long long int sum[100005];
    memset();
    while(~)
    {
        n=getchar();x=getchar();y=getchar();
        l=0;
        r=n;
        one=0;zero=0;m=2;
        ans=0;
        for(l=1;l<r;l++)
        {
            if(str[m-1]==1)
                k=1;y=0;
            else
                k=0;y=1;
            if(x>y)
                zero+=1;
            else
                one+=1;
            m++;
            if(str[m]==1)
                k=1;
            if(y>x)
                one--;
            else
                zero--;
        }
        sum[0]=0;
        for(i=1;i<=n;i++)
            sum[i]=sum[i-1]+abc1[i-1];
        printf("%d\n",sum[n]);
        if(str[0]<'0'||str[0]>'9')
        {
            ans=0;
            for(i=0;i<n;i++)
            {
                if(str[i]>(l-1))
                    break;
            }
            if(str[i]>(r-1))
                break;
            if(str[i]>(l-1))
                ans+=sum[i];
            if(str[i]<(r+1))
                ans-=sum[i];
        }
        else
        {
            a=i-(l-1);
            b=i-(r+1);
            ans=a*b/2;
        }
        printf("%d\n",ans);
    }
    printf("%d\n",ans);
}
while(0)
{
    l=getchar();r=getchar();
    ans=0;
    if(r-l>0)
        ans+=sum[r]-sum[l-1];
    else
    {
        a=i-(l-1);
        b=i-(r+1);
        ans=a*b/2;
    }
}
printf("%d\n",ans);
}
return 0;
}
```

**Sample Input**

```
1
8 2 3
0 1 1 0 0 0 0
1 4
2 4
5 8
5 5
7
```

**Sample Output**

```
8
```

**Result**

Thus, Program " Substring " has been successfully executed

**Q. Symbols Filter**

Ganga found a diary, she can't understand what is written in it. Because the letters are mingled with special symbols. She needs to filter those letters to read that diary. Can you help her?

**Source Code**

```
#include <stdio.h>
int main()
{
    char line[150];

    int i,j;

    scanf("%s",line);

    for(i=0;line[i]!='\0';++i)
    {
        while(!((line[i]>='a' && line[i]<='z')||(line[i]>='A' && line[i]<='Z')||line[i]=='\0'))
        {
            for(j=i;line[j]!='\0';++)
            {
                line[j]=line[j+1];
            }
            line[j]='\0';
        }
    }
    printf("%s",line);
    return 0;
}
```

**Sample Input**

pass@word

**Sample Output**

password

**Result**

Thus, Program " **Symbols Filter** " has been successfully executed

**Q. Sum of even and odd numbers**

Write a program to find the sum of even and odd numbers in an array.

Input Format:

Input consists of  $n+1$  integers. The first integer corresponds to  $n$ , the size of the array. The next  $n$  integers correspond to the elements in the array. Assume that the maximum value of  $n$  is 15.

Output Format:

Refer sample output for details

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,i,even,odd;
    scanf("%d",&a);
    for(i=0;i<a;i++)
    {
        scanf("%d",&b);
        if(b%2==0)
        {
            even=even+b;
        }
        else
        {
            odd=odd+b;
        }
    }
    printf("even=%d",even);
    printf("\nodd=%d",odd);
    return 0;
}
```

**Sample Input**

5  
2 3 6 8 -1

**Sample Output**

even=16  
odd=2

**Result**

Thus, Program " Sum of even and odd numbers " has been successfully executed

**Q. Catch the Second..**

A bag contains set of integers and it is discovered by Ram. There is sheet in that bag. The person who found the bag should find the second largest number from the set of numbers. If the rightly found the will get a gold.. If you wish get the prize you can also try...

**Source Code**

```
#include <stdio.h>
int main()
{
    int a[50],size,i,j=0,big,sbig;
    scanf("%d",&size);
    for(i=0;i<size;i++)
        scanf("%d",&a[i]);
    big=a[0];
    for(i=1;i<size;i++)
    {
        if(big<a[i])
        {
            big=a[i];
            j=i;
        }
    }
    sbig=a[size-j-1];
    for(i=1;i<size;i++)
    {
        if(sbig<a[i]&&j!=i)
            sbig=a[i];
    }
    printf("%d",sbig);

    return 0;
}
```

**Sample Input**

5  
5 3 1 2 6

**Sample Output**

5

**Result**

Thus, Program " **Catch the Second..** " has been successfully executed

**Q. Array Deletion**

Write a program to delete an element from the array.

**Input and Output Format:**

Assume that the maximum number of elements in the array is 20.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Input:**

1. The number of inputs to be entered by the user

2. The array elements

3. The array index to be deleted

**Example 1:**

**Input :**

4  
11 12 13 14

**Output 1:**

Array after deletion is

11 13 14

**Explanation:**

Array index starts from 0 the user entered array index as 1 so the number 12 is deleted.

**int main()**

{  
    int a[100],n,pos;

    scanf("%d",&n);

    for (i=0;j<n;i++)

{  
        scanf("%d",&a[i]);

    }

    scanf("%d",&pos);

    for(i=pos;j<n-1;j++)

{  
        a[j]=a[i+1];

    }

n=n-1;

printf("Array after deletion is\n");

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

{  
    printf("%d ",a[i]);

}

return 0;

}

**Sample Input**

5  
14 25 13 34 45  
4

**Sample Output**

Array after deletion is  
14 25 13 34

**Result**

Thus, Program " **Array Deletion** " has been successfully executed

### Q. Devu and friendship testing

Devu has  $n$  weird friends. Its his birthday today, so they thought that this is the best occasion for testing their friendship with him. They put up conditions before Devu that they will break the friendship unless he gives them a grand party on their chosen day. Formally,  $i$ th friend will break his friendship if he does not receive a grand party on  $i$ th day.

Devu despite being as rich as Gatsby, is quite frugal and can give at most one grand party daily. Also, he wants to invite only one person in a party. So he just wonders what is the maximum number of friendships he can save. Please help Devu in this tough task !!

#### Input

The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.

First line will contain a single integer denoting  $n$ .

Second line will contain  $n$  space separated integers where  $i$ th integer corresponds to the day  $i$ th as given in the problem.

#### Output

Print a single line corresponding to the answer of the problem.

#### Constraints

$1 \leq T \leq 10^4$   
 $1 \leq n \leq 50^4$   
 $1 \leq d_i \leq 100$

#### Explanation

Example case 1. Devu can give party to second friend on day 2 and first friend on day 3, so he can save both his friendships.

Example case 2. Both the friends want a party on day 1, and as the Devu can not afford more than one party a day, so he can save only one of the friendships, so answer is 1.

### Source Code

```
#include<stdio.h>
int main()
{ int n,i,l,n,sum[100],p;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&n);
    int a;
    for(i=0;i<100;i++)
      sum[i]=0;

    for(i=0;i<n;i++)
    {
      scanf("%d",&a);
      sum[(a-1)]=1;
    }
    p=0;
    for(i=0;i<100;i++)
    {
      if(sum[i]==1)
        p=(p+1);
    }
    printf("%d\n",p);
  }
  return 0;
}
```

### Sample Input

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

### Sample Output

```
2
1
```

### Result

Thus, Program " **Devu and friendship testing** " has been successfully executed

**Q. TRIANGULAR NUMBERS**

"A triangular number is the number of dots in an equilateral triangle uniformly filled with dots. For example, three dots can be arranged in a triangle; thus three is a triangular number. The n-th triangular number is the number of dots in a triangle with n dots on a side. You can learn more about these numbers from Wikipedia ([http://en.wikipedia.org/wiki/Triangular\\_number](http://en.wikipedia.org/wiki/Triangular_number)). Your task is to find out if a given integer is a triangular number.

Input:

The first line contains the single number n (1. Output:  
If the given integer is a triangular number output YES, otherwise output NO.

**Source Code**

```
#include <stdio.h>
int triangle(int num)
{
    if(num<0)
        return 0;
    int n,sum;
    for(n=1;sum<num;n++)
    {
        sum=sum+n;
        if(sum==num)
            return 1;
    }
    return 0;
}
int main()
{
    int n;
    scanf("%d",&n);
    if(triangle(n))
        printf("YES");
    else
        printf("NO");
    return 0;
}
```

**Sample Input**

1

**Sample Output**

YES

**Result**

Thus, Program " **TRIANGULAR NUMBERS** " has been successfully executed

**Q. PRINT 3**

Write a C program to print all numbers between a and b ( a and b inclusive) using a while loop.

Input format:

Input consists of 2 integers. The first integer corresponds to a and the second integer corresponds to b . Assume a>=b.

Output format:

Refer sample input and output for formatting specifications.

**Source Code**

```
#include <stdio.h>
int main() {
    int a,b;
    scanf("%d",&a);
    scanf("%d",&b);
    while(b<=a)
    {
        printf("%d\n",a);
        a--;
    }
    return 0;
}
```

**Sample Input**

```
10
4
```

**Sample Output**

```
10
9
8
7
6
5
4
```

**Result**

Thus, Program " **PRINT 3** " has been successfully executed

**Q. Blood Bank**

A team from SRM Rotaract club had planned to conduct rally to create awareness among the chennai people to denote blood. They conducted the rally successfully. Many of the chennai people realise it and came forward to donate their blood to near by blood bank. The eligibility criteria for donating blood is people should be above 18 and his/her weight should be above 40. There was a huge crowd and staff in blood bank found it difficult to manage the crowd. So they decided to keep a system and ask the people to enter their age and weigh in a system. If a person is eligible he/she will be allowed inside.

Write a C program and feed it to the system to find whether a person is eligible or not.

Input format:

Input consists of 2 integers which corresponds to age and weight of a person respectively.

Output Format:

Display whether person is eligible or not.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    if(a>18 && b>40)
        printf("Eligible to donate");
    else
        printf("Not Eligible to donate");
    return 0;
}
```

**Sample Input**

19  
41

**Sample Output**

Eligible to donate

**Result**

Thus, Program " **Blood Bank** " has been successfully executed

**Q. FIND ALL DUPLICATES IN ARRAY**

Given an array of integers,  $1 < a[i] < n$ . let N be the number of test cases and n be the number of input integers. Some elements appear twice and others appear once. Find all the elements that appear twice in this array.  
Input format: Get n number of inputs of size of an array in separated strings.  
Output format: Print the duplicate integer in the given list.  
Explanation: From the input, search for the repeated value. If the repeated value is found print it as an output in sequential order.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,i,j;
    scanf("%d",&a);
    int arr[a];
    for(i=0;i<a;i++)
        scanf("%d",&arr[i]);
    for(i=0;i<a;i++)
    {
        for(j=i+1;j<a;j++)
        {
            if(arr[i]==arr[j])
                printf("%d ",arr[i]);
        }
    }
    return 0;
}
```

**Sample Input**

```
7
1 2 8 5 7 5 2
```

**Sample Output**

```
2 5
```

**Result**

Thus, Program " **FIND ALL DUPLICATES IN ARRAY** " has been successfully executed

**Q. Lucy**

Lucy is celebrating her 15th birthday. Her father promised her that he will buy her a new computer on her birthday if she solves the question asked by him. He asks Lucy to find whether the year on which she had born is leap year or not. Help her to solve this puzzle so that she celebrates her birthday happily. If her birth year is 2016 and it is a leap year display 2016 is a leap year.? Else display 2016 is not a leap year and check with other leap year conditions

**Source Code**

```
#include <stdio.h>
int main()
{
    int year;
    scanf("%d",&year);
    if(year%4 == 0)
    {
        if(year%100 == 0)
        {
            if(year%400 == 0)
            {
                printf("%d is a leap year",year);
            }
            else
            {
                printf("%d is not a leap year",year);
            }
        }
        else
        {
            printf("%d is a leap year",year);
        }
    }
    else
    {
        printf("%d is a leap year",year);
    }
    return 0;
}
```

**Sample Input**

1900

**Sample Output**

1900 is not a leap year

**Result**

Thus, Program " Lucy " has been successfully executed

### Q. Lab seating arrangement

There are 2 Programming Labs . Each with a seating capacity of 90. There are 240 students with registration numbers from 1 to 240. All 240 students cannot be accommodated in the labs at the same time. It has been decided to conduct theory class for 60 students every week. It has been planned to conduct theory classes for all students with register number being a multiple of 4. Students with registration number from 1 to 120 with registration number not a multiple of 4 need to be seated in programming lab I and students with registration numbers from 121 to 240 with registration numbers not a multiple of 4 need to be seated in programming lab II.

Given the registration number of student, write a C program to specify the lab or hall in which student need to be seated.

Input Format:

Input consists of 1 integer which corresponds to the registration number of the student.

Output format:

Output consists of string "Lab 1" or "Lab 2" or "Theory session" or "Incorrect Register Number"

Refer sample input and output for further formatting specifications.

Example 1:

99  
Output=Lab 1

Example 2:

241  
Output=Incorrect Register Number

### Source Code

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%4==0&&a!=0)
    {
        printf("Theory session");
    }
    else if(a<120&&a>1)
    {
        printf("Lab 1");
    }
    else if(a<240&&a>120)
    {
        printf("Lab 2");
    }
    else if(a>240||a<1)
    {
        printf("Incorrect Register Number");
    }
    return 0;
}
```

### Sample Input

16

### Sample Output

Theory session

### Result

Thus, Program " Lab seating arrangement " has been successfully executed

**Q. Cyclic Number**

Write a C program to swap elements in cyclic order using call by reference.

**Source Code**

```
#include <stdio.h>
void Cycle(int *a,int *b,int *c);
int main()
{
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    Cycle(&a,&b,&c);
    printf("%d\n%d\n%d\n",a,b,c);
    return 0;
}
void Cycle(int *a,int *b,int *c)
{
    int temp;
    temp=*b;
    *b=*a;
    *a=*c;
    *c=temp;
}
```

**Sample Input**

1 2 3

**Sample Output**

3  
1  
2

**Result**

Thus, Program " **Cyclic Number** " has been successfully executed

### Q. Pogo Stick Jump

Raju lives in a colony. On his 9th birthday, his father gift him a Pogo Stick. He is so excited to play with pogo stick. The pogo stick moves one unit per jump. He wanders around his house jumping with pogo sticks. He wants to show the pogo stick to his friends and decide to go using pogo sticks. Write a program to find number of jumps needed to reach his friends house. Assume that Rajus house is in the location (3,4).

Input and Output Format:  
Input consists of two integers x, y. The x and y corresponds to x and y coordinates of his friends house.  
Output is an integer - the number of jumps he needs to reach his friends house.

### Source Code

```
#include <stdio.h>
int main()
{
    int x,y,i,j;
    scanf("%d %d",&x,&y);
    i=x-3;
    j=y-4;
    printf("Raju needs %d jumps",j);
    return 0;
}
```

### Sample Input

5 6

### Sample Output

Raju needs 2 jumps

### Result

Thus, Program " Pogo Stick Jump " has been successfully executed

### Q. Month Calculator

The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display 28 or 29 days for February so that leap years are addressed.

The program should get input from Jan to Dec.

If the input is from May to other months then display error messages as "Invalid"

### Source Code

```
#include <stdio.h>
int main()
{
    char
    month[100],a[]="Jan",b[]="Feb",c[]="Mar",d[]="Apr",e[]="May",f[]="Jun",g[]="Jul",h[]="Aug",i[]="Sep",j[]="Oct",k[]="Nov";
    scanf("%s",month);
    if(strcmp(month,a)==0)
        printf("31");
    else if(strcmp(month,b)==0)
        printf("28 or 29");
    else if(strcmp(month,c)==0)
        printf("31");
    else if(strcmp(month,d)==0)
        printf("30");
    else if(strcmp(month,e)==0)
        printf("31");
    else if(strcmp(month,f)==0)
        printf("30");
    else if(strcmp(month,g)==0)
        printf("31");
    else if(strcmp(month,h)==0)
        printf("31");
    else if(strcmp(month,i)==0)
        printf("30");
    else if(strcmp(month,j)==0)
        printf("31");
    else if(strcmp(month,k)==0)
        printf("30");
    else
        printf("31");
    return 0;
}
```

### Sample Input

Jun

### Sample Output

30

### Result

Thus, Program " Month Calculator " has been successfully executed

### Q. triangle height

Chef belongs to a very rich family which owns many gold mines. Today, he brought N gold coins and decided to form a triangle using these coins. Isn't it strange?

Chef has an unusual way of forming a triangle using gold coins, which is described as follows:

He puts 1 coin in the 1st row.

then puts 2 coins in the 2nd row.

then puts 3 coins in the 3rd row.

and so on as shown in the given figure.

Chef is interested in forming a triangle with maximum possible height using at most N coins. Can you tell him the maximum possible height of the triangle?

### Source Code

```
#include <stdio.h>
int main()
{
    int T,i,j;
    scanf("%d",&T);
    long int n[T];
    for(i=0;i<T;i++)
    {
        scanf("%li",&n[i]);
    }
    for(i=0;i<T;i++)
    {
        j=i;
        while(n[i]>=0)
        {
            n[i]=n[i]-j;
            j++;
        }
    }
    printf("%d\n",j-2);
}
return 0;
}
```

### Sample Input

```
3
3
5
7
```

### Sample Output

```
2
2
3
```

### Result

Thus, Program " triangle height " has been successfully executed

### Q. Greedy puppy

Tuzik is a little dog. But despite the fact he is still a puppy he already knows about the pretty things that coins are. He knows that for every coin he can get very tasty bone from his master. He believes that some day he will find a treasure and have loads of bones.

And finally he found something interesting. A wooden chest containing N coins. But as you should remember, Tuzik is just a little dog, and so he can't open it by himself. Actually the only thing he can really do is barking. He can use his barking to attract nearby people and seek their help. He can set the loudness of his barking very precisely, and therefore you can choose that he can choose to call any number of people, from a minimum of 1, to a maximum of K.

When people come and open the chest they divide all the coins between them in such a way that everyone will get the same amount of coins and this amount is maximal possible. If some coins are not used they will leave it on the ground and Tuzik will take them after they go away. Since Tuzik is clearly not a fool, he understands that his profit depends on the number of people he can get.

Input:

The first line of the input contains an integer T denoting the number of test cases. Each of next T lines contains 2 space-separated integers: N and K, for this test case.

Output:

For each test case output one integer - the maximum possible number of coins Tuzik can get.

Constraints

$1 \leq N \leq 50$

$1 \leq K \leq 105$

In the first example he should call two people. Each of them will take 2 coins and they will leave 1 coin for Tuzik.

In the second example he should call 3 people

EXPLANATION :

As the size of the inputs in the problem is very small ( in the order of  $10^2$  ), we can simply use brute-forcing ( Brute Forcing ) and test every possible case and print the value of the maximum coins that the dog gets.

As the dog only gets what remains after equal splitting , this can be calculated by using the modulo " % " ( Modulo ) operation .

Example :

For 2nd test case : { 11 3 } .

He can call maximum of 3 people

If he calls only 1 person , he will take all the coins , leaving him with nothing

If he calls 2 people , he will be left with only 1 coin as the two people take 5 coins each.

If he calls 3 people , each one will take 3 coins , leaving him with 2 coins

So the best case for him is to call 3 people so he can get 2 coins.

The input format is :

An integer T to represent the number of test cases. ( number of times the user will provide distinct values of N and K )

The result will be biggest modulo value that a number gives with N , such that the number is less than K.

### Source Code

```
#include <stdio.h>
int main()
{
    int ln,k;
    scanf("%d",&ln);
    int i=0,max=0;
    for(i=0;i<ln;i++)
    {
        scanf("%d%d",&n,&k);
        for(j=1;j<=k;j++)
        {
            if((n%j)>max)
            {
                max=n%j;
            }
        }
        printf("%d\n",max);
        max=0;
    }
    return 0;
}
```

### Sample Input

```
2
5 2
11 3
```

### Sample Output

```
1
2
```

### Result

Thus, Program " Greedy puppy " has been successfully executed

#### **Q. Null or Not**

George doesn't have a clear vision to understand whether the number is null or not. If we write a programming language means, it is very easy to understand. Help George to write a C program to check whether the given number is null or not.

#### **Source Code**

```
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a==0)
        printf("NULL");
    else
        printf("NOT NULL");
    return 0;
}
```

#### **Sample Input**

12

#### **Sample Output**

NOT NULL

#### **Result**

Thus, Program " Null or Not " has been successfully executed

**Q. Calculating SP**

A man buys a cycle for Rs. A and sells its a loss of B% . What is the selling price of the cycle?

Write a C program to compute the selling price

Input format:

The first input is an integer which corresponds to A . The second input is an integer which corresponds to B.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b;
    float loss,tot;
    scanf("%d%d",&a,&b);
    if(a==0)
    {
        printf("%d",a);
    }
    else
    {
        loss=a*(b*0.01);
        tot=a-loss;
        printf("The selling price of the cycle is Rs=%.2f",tot);
    }
    return 0;
}
```

**Sample Input**

1400  
15

**Sample Output**

The selling price of the cycle is Rs=1190.00

**Result**

Thus, Program " **Calculating SP** " has been successfully executed

**Q. Nth FIBO**

John played with jacob.John wanted to test how soon jacob will answer for his question.He told him one number.The job of john is to add the numbers 0 and 1 initially and to add that output with 1.Now Jacob gets another output.jacob has to add this current output with previous output .This action has to be repeated upto certain times.After this Jacob needs to write those outputs as a series starts from 0,1 and to find out the n th number of that series what john told.Write a code for this.

**Source Code**

```
#include <stdio.h>
int main()
{
    int num;
    int result;
    scanf("%d",&num);
    if(num<0)
    {
    }
    else
    {
        result = fibo(num);
        printf("%d\n",result);
    }
    return 0;
}
int fibo(int num)
{
    if(num == 0)
    {
        return 0;
    }
    else if (num == 1)
    {
        return 1;
    }
    else
    {
        return(fibo(num-1)+fibo(num-2));
    }
}
```

**Sample Input**

8

**Sample Output**

21

**Result**

Thus, Program " **Nth FIBO** " has been successfully executed

### Q. Find the Year

Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years.

The rules for determining whether or not a year is a leap year follow:

- Any year that is divisible by 400 is a leap year.
- Of the remaining years, any year that is divisible by 100 is not a leap year.
- Of the remaining years, any year that is divisible by 4 is a leap year.
- All other years are not leap years.

Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

### Source Code

```
#include <stdio.h>
#include<math.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%4==0)
        printf("Yes");
    else
        printf("No");
    return 0;
}
```

### Sample Input

2016

### Sample Output

Yes

### Result

Thus, Program " **Find the Year** " has been successfully executed

**Q. GCD and LCM**

Two integers A and B are the inputs. Write a program to find GCD and LCM of A and B.  
Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer A and B.

Output

Display the GCD and LCM of A and B separated by space respectively.  
Constraints

$1 \leq T \leq 1000$   
 $1 \leq A, B \leq 1000000$

**Source Code**

```
#include <stdio.h>
int main()
{int m,a,b,n,d,r,gcd,lcm;
scanf("%d",&m);
while(m--)
{
    scanf("%d %d",&a,&b);
    if(a>b)
    {
        n=a;
        d=b;
    }
    else
    {
        d=a;
        n=b;
    }
    r=n%d;
    while(r!=0)
    {
        n=d;
        d=r;
        r=n%d;
    }
    gcd=d;
    lcm=(a*b)/gcd;
    printf("%d %d\n",gcd,lcm);
}
return 0;
}
```

**Sample Input**

```
5
2 3
2 4
3 5
4 6
7 8
```

**Sample Output**

```
1 6
2 4
1 15
2 12
1 56
```

**Result**

Thus, Program " **GCD and LCM** " has been successfully executed

**Q. GCD and LCM**

Two integers A and B are the inputs. Write a program to find GCD and LCM of A and B.  
Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer A and B.

Output

Display the GCD and LCM of A and B separated by space respectively.  
Constraints

$1 \leq T \leq 1000$   
 $1 \leq A, B \leq 1000000$

**Source Code**

```
#include <stdio.h>
int main()
{int m,a,b,n,d,r,gcd,lcm;
scanf("%d",&m);
while(m--)
{
    scanf("%d %d",&a,&b);
    if(a>b)
    {
        n=a;
        d=b;
    }
    else
    {
        d=a;
        n=b;
    }
    r=n%d;
    while(r!=0)
    {
        n=d;
        d=r;
        r=n%d;
    }
    gcd=d;
    lcm=(a*b)/gcd;
    printf("%d %d\n",gcd,lcm);
}
return 0;
}
```

**Sample Input**

```
5
2 3
2 4
3 5
4 6
7 8
```

**Sample Output**

```
1 6
2 4
1 15
2 12
1 56
```

**Result**

Thus, Program " **GCD and LCM** " has been successfully executed

**Q. Star formation**

Pyramid Star Formation

**Source Code**

```
#include <stdio.h>
int main()
{
    int i, j, rows;
    scanf("%d",&rows);
    for (i = 1; i <= rows; i++) {
        for (j = 0; j < i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

**Sample Input**

5

**Sample Output**

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

**Result**

Thus, Program " **Star formation** " has been successfully executed

**Q. Favorite Sequence**

Lancy has a sequence of N numbers. He like a sequence better if the sequence contains his favorite sequence as a substring.

Given the sequence and his favorite sequence(F) check whether the favorite sequence is contained in the sequence

Input  
The first line will contain the number of test cases and are followed by the cases

Each test case consists of four lines: The length of the sequence, N, the length of F and the sequence F

Print "Yes" if the sequence contains the favourite sequence int it otherwise print "No"

Constraints

$1 \leq T \leq 10$

**Source Code**

```
#include <stdio.h>
int main()
{
    int t;
    scanf("%d", &t);
    while(t--)
    {
        int n1,n2,f=0,k=0;
        scanf("%d", &n1);
        int a[n1];
        for(i=0;i<n1;i++)
        {
            scanf("%d", &a[i]);
        }
        scanf("%d", &n2);
        int b[n2];
        for(i=0;i<n2;i++)
        {
            scanf("%d", &b[i]);
        }
        for(i=0;i<n1&&k<n2;i++)
        {
            if(a[i]==b[k])
            {
                f++;
                k++;
            }
            if(f==n2)
            break;
        }
    }
    if(f>=n2)
    printf("Yes");
    else
    printf("No");
}
return 0;
}
```

**Sample Input**

```
2
6
1 2 3 4 5 6
3
2 3 4
6
22 5 6 33 14
2
4 15
```

**Sample Output**

```
Yes
No
```

**Result**

Thus, Program " Favorite Sequence " has been successfully executed

**Q. Division of numbers(Integer data type)**

Jennys home work for Fourth day is to find Division of two numbers, help jenny to solve the problem.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,c;
    scanf("%d",&a);
    scanf("%d",&b);
    c=a/b;
    printf("The division of two number is:%d",c);
    return 0;
}
```

**Sample Input**

```
6
2
```

**Sample Output**

The division of two number is:3

**Result**

Thus, Program " **Division of numbers(Integer data type)** " has been successfully executed

**Q. Your Name is Mine**

In an attempt to control the rise in population, Archer was asked to come up with a plan. This time he is targeting marriages. Archer, being as intelligent as he is, came up with the following plan:

A man with name M is allowed to marry a woman with name W, only if M is a subsequence of W or W is a subsequence of M.

A is said to be a subsequence of B, if A can be obtained by deleting some elements of B without changing the order of the remaining elements.

Your task is to determine whether a couple is allowed to marry or not, according to Archers rule.

Input  
The first line contains an integer T, the number of test cases. T test cases follow. Each test case contains two space separated strings M and W.

Output  
For each test case print "YES" if they are allowed to marry, else print "NO". (quotes are meant for clarity, please dont print them)

Constraints  
1 <= T <= 100  
1 <= |M|, |W| <= 25000 (A denotes the length of the string A.)

All names consist of lowercase English letters only.

**Source Code**

```
#include <stdio.h>
#include <iostream.h>
#include <string.h>
#define p 25000

int main()
{
    char s1[p],s2[p];
    int g=0,i,t,k=0;
    scanf("%s",s1);
    scanf("%s",s2);
    for(i=0;s2[i]!='\0';i++)
    {
        if(s2[i]==s1[g])
        {
            g++;
        }
    }
    for(i=0;s1[i]!='\0' && s2[i]!='\0';i++)
    {
        if(s1[i]==s2[i])
            h++;
    }
    int l=strlen(s1);
    int l1=strlen(s2);
    if(l==g || l1==h)
        printf("YES\n");
    else
        printf("NO\n");
}
return 0;
}
```

**Sample Input**

```
3
john johanna
ira ira
kaya jayla
```

**Sample Output**

```
YES
YES
NO
```

**Result**

Thus, Program " Your Name is Mine " has been successfully executed

**Q. SMS Languages**

SMS language or textese (also known as txt-speak, txteze, chatspeak, txt, txtpk, txtk, txt, texting language, txt lingo, SMSish, txtslang, or txt talk) is a term for the abbreviations and slang commonly used with mobile phone text messaging.

Some of the abbreviations used are

s for yes  
u for you  
2day for today  
y for why

Many grandpa's have started sending SMSes to their grand children. But they are not familiar with the SMS lingo.

Input Format:

Input consists of a single string. Assume that the maximum length of the string is 200 and all letters are in lower-case.

Output Format:

Output consists of a single string.

Sample Input 1:

where were u yesterday?

Sample Output 1:

where were u sterday?

Sample Input 2:

why is today a working day for you?

Sample Output 2:

y is 2day a working day for u?

**Source Code**

```
#include <stdio.h>
#include <string.h>

void replace(char *s1)
{
    char r[4][10]={"yes","you","today","why"};
    char rep[4][5]={"s","u","2day","y"};
    char buff[500];
    int i;
    char *ch;
    for(i=0;i<4;i++)
        if((ch=strstr(s1,r[i])))
            strcpy(buff,s1,ch-s1);
    buff[500]=0;
    sprintf(buff+ch-s1,"%s%s",rep[i],ch+strlen(r[i]));
    s1[0]=0;
    strcpy(s1,buff);
    return replace(s1);
}

int main()
{
    char s1[100];
    fgets(s1,100,stdin);
    replaces(s1);
    printf("%s",s1);
    return 0;
}
```

**Sample Input**

where were you yesterday

**Sample Output**

where were u sterday

**Result**

Thus, Program " **SMS Languages** " has been successfully executed

**Q. read data on a regular basis**

"Alexey is trying to develop a program for a very simple microcontroller. It makes readings from various sensors over time, and these readings must happen at specific regular times. Unfortunately, if two of these readings are taken from different sensors at the same time, then the reading from sensor  $i$  will be "frozen" to  $N_i$ , the reading from sensor  $i$  will occur every  $A_i$  milliseconds after the first reading from sensor  $i$ . Each reading takes exactly one millisecond on Alexey's microcontroller. Alexey wants to know when the microcontroller will freeze after it is turned on.

The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.

The second line of the input contains a single integer  $N$  denoting the number of sensors.

The third line of the input contains integers  $A_1, A_2, \dots, A_N$  denoting frequency of measurements. Namely, sensor  $i$  will be read every  $A_i$  milliseconds with the first reading occurring at  $A_i$  milliseconds after the first line.

For each test case, output a single line containing the number of milliseconds until the microcontroller freezes.

Constraints

$1 \leq N \leq 10$

$2 \leq N < 500$

$1 < A_i < 10^8$

**Source Code**

```
#include <stdio.h>
int gcd(int u,int v)
{
    long int t;
    while(t>0)
    {
        if(u<v)
        {
            t=u;
            u=v;
            v=t;
        }
        u=u-v;
    }
    ans=(a%"b)/v;
}
return ans;
}
int main()
{
    int test,z;
    scanf("%d",&test);
    for(z=0;z<test;z++)
    {
        int n,i,j;
        scanf("%d",&n);
        int num[n],ans=1000000000;
        for(i=0;i<n;i++)
        {
            scanf("%d",&num[i]);
        }
        for(i=0;i<n;i++)
        {
            for(j=i+1;j<n;j++)
            {
                long int temp=gcd(num[i],num[j]);
                if(temp<ans)
                    ans=temp;
            }
        }
        printf("%d\n",ans);
    }
    return 0;
}
```

**Sample Input**

```
3
3
3
2 3 5
4
1 8 11
4
4 4 5 6
```

**Sample Output**

```
6
6
7
4
```

**Result**

Thus, Program " **read data on a regular basis** " has been successfully executed

**Q. Print prime numbers**

Ravi is poor in mathematics; his teacher wants him to improve his problem solving skills. So he gives a number to Ravi and asked him to find the prime numbers between 2 and the given number.

**Source Code**

```
#include <stdio.h>
int main()
{
    int n,i,fact,j;
    scanf("%d",&n);
    for(i=3;i<n;j++)
    {
        fact=0;
        for(j=1;j<n;j++)
        {
            if(i%j==0)
                fact++;
        }
        if(fact==2)
            printf("%d ",i);
    }
    return 0;
}
```

**Sample Input**

11

**Sample Output**

3 5 7

**Result**

Thus, Program " **Print prime numbers** " has been successfully executed

**Q. Push Ups withBlaze**

At UAB football games, Blaze does push ups after each Blazer score. After the first Blazer touchdown (and point after), Blaze does 7 push ups. After the second touchdown and point after, the score is now 14 and Blaze does 14 push ups.

Write a program that calculates how many total push ups Blaze does during the whole game. Assume that only 7 point touchdowns (including the point after) occur. Prompt for the final score and print out how many push ups Blaze has done.

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,i;
    scanf("%d",&a);
    for(i=7;i<=a;i+=7)
        b=b+i;
    printf("%d",b);
    return 0;
}
```

**Sample Input**

21

**Sample Output**

42

**Result**

Thus, Program " **Push Ups withBlaze** " has been successfully executed

### Q. Sita Out

Ms. Sita, the faculty handling programming lab for you is very strict. Your seniors have told you that she will not allow you to enter the week's lab if you have not completed atleast half the number of problems given last week.

Many of you didn't understand this statement and so they requested the good programmers from your batch to write a program to find whether a student will be allowed into a week's lab given the number of problems given last week and the number of problems solved by the student in that week.

Can you help in writing this program?

Input Format:

Input consists of 2 integers. The first integer corresponds to the number of problems given and the second integer corresponds to the number of problems solved.

Output Format:

Output consists of the string IN or OUT.

Refer sample input and output for further formatting specifications.

### Source Code

```
#include <stdio.h>
int main()
{
    int mincount, maxcount,c;
    scanf("%d%d",&maxcount,&mincount);
    c = maxcount/2;
    if(mincount<c)
        printf("OUT");
    else
        printf("IN");
    return 0;
}
```

### Sample Input

8 3

### Sample Output

OUT

### Result

Thus, Program " **Sita Out** " has been successfully executed

#### **Q. CHANGE IT**

Mani Working as a professor in ABC college, have to get students three subjects points. So he planned to do one program to implement

Input

3 3 5

Output

3 3 5

#### **Source Code**

```
#include <stdio.h>
struct display
{
    int a,b,c;
}s;

int main()
{
    struct display s;
    scanf("%d%d%d",&s.a,&s.b,&s.c);
    printf("%d %d %d",s.a,s.b,s.c);
    return 0;
}
```

#### **Sample Input**

3 3 5

#### **Sample Output**

3 3 5

#### **Result**

Thus, Program " **CHANGE IT** " has been successfully executed

**Q. IO 30**

Write a program to read the address of a user. Display the result by breaking it into multiple lines

Input and Output Format:

Refer sample input and output for formatting specification.

All float values are displayed correct to 2 decimal places.

All text in bold corresponds to input and the rest corresponds to output

**Source Code**

```
#include <stdio.h>

int main()
{
    int a,b;
    char c[20];
    scanf("%d",&a);
    scanf("%s",c);
    scanf("%d",&b);
    printf("Address");
    printf("\nHouse Number:%d",a);
    printf("\nArea:%s",c);
    printf("\nPin Code:%d",b);
    return 0;
}
```

**Sample Input**

3489  
SRMChennai  
603203

**Sample Output**

Address  
House Number:3489  
Area:SRMChennai  
Pin Code:603203

**Result**

Thus, Program " **IO 30** " has been successfully executed

**Q. IO 19**

Heera is a cute little gal of age 5 years old, she had one piggybank to save money (coins). Her piggy bank got full and she was excited to her savings money. So, she classified the coins of its followings types , Rs.10 coins, Rs. 5 Coins , Rs .2 Coins, Rs. 1 Coins. Now Please help her to calculate the total amount she saved?

Input :

total number of coins collected on every classified type respectively( Rs..10 Coins, Rs. 5 Coins, Rs. 2 coins, Rs.1 Coins)

**Source Code**

```
#include <stdio.h>
int main()
{
    int a,b,c,d;
    float amount;
    scanf("%d\n",&a);
    scanf("%d\n",&b);
    scanf("%d\n",&c);
    scanf("%d\n",&d);
    amount=(float)(a*10)+(b*5)+(c*2)+(d*1);
    printf("Total amount in the piggybank=%0.2f",amount);
    return 0;
}
```

**Sample Input**

```
10
23
43
6
```

**Sample Output**

Total amount in the piggybank=307.00

**Result**

Thus, Program " **IO 19** " has been successfully executed