

## TCS NINJA – CODING SET 1

### PROGRAM 1

Program to get input and print it. Get the input using input statement and print it using output statement.

#### Test Case 1

##### Input

Hai

##### Output

Hai

#### Test Case 2

##### Input

Hello

##### Output

Hello

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    char str[20];
    cin>>str;
    cout<<str;
    return 0;
}
```

## PROGRAM 2

Write a program to get the values from the user and print the same to the output screen. Program to get the value as an input using scanf function and print the same values as an output to the screen using printf functions

### Test Case 1

#### Input

10 hai 1.6 s

#### Output

10 hai 1.600000 s

### C Program

```
#include <stdio.h>
int main()
{
    int dec;
    char str[10];
    char ch;
    float pi;
    scanf("%d %s %f %c", &dec, str, &pi, &ch);
    printf("%d %s %f %c", dec, str, pi, ch);
    return 0;
}
```

### C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    int dec;
    char str[10];
    char ch;
    float pi;
    cin>>dec>>str>>pi>>ch;
    printf("%d %s %f %c", dec, str, pi, ch);
    return 0;
}
```

### PROGRAM 3

Booka is an alien. He couldn't understand how to measure days, weeks, months and years. Make Booka understand what is meant by days, weeks, months and years. Teach him about the conversion of days into years, months and weeks using a program.

**INPUT FORMAT:** Input consists of an integer which corresponds to the number of days.

**OUTPUT FORMAT:** The output consists of three integers. The first integer corresponds to the total years. The second integer corresponds to the total weeks. The third integer corresponds to the total days.

#### Test Case 1

##### Input

373

##### Output

1

1

1

#### Test Case 2

##### Input

366

##### Output

1

0

0

#### Test Case 3

##### Input

367

##### Output

1

0

0

### C++ Program

```
#include<iostream>
```

```
#include<math.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,y,w,d;
```

```
    cin>>a;
```

```
    y= (a/365);
```

```
cout<<y<<"\n";  
w=(a-365)/7;  
cout<<w<<"\n";  
d=w/1;  
cout<<d;  
return 0;  
}
```

## PROGRAM 4

Ajay was a student who is interested in coding. One fine day Ajay's professor challenged all the students to calculate the power of 2 without using Arithmetic operators. Ajay want's to solve this question but he don't know which is the right operator to be selected. Help Ajay by writing a C++ program to calculate the power of 2 without using arithmetic operator.

### Test Case 1

#### Input

3

#### Output

8

### Test Case 2

#### Input

0

#### Output

1

### C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    cout<<(1<<n);
    return 0;
}
```

## PROGRAM 5

Your task is to write a program to find whether the given year is Leap year or Not.

### Test Case 1

#### Input

2000

#### Output

LEAP YEAR

### Test Case 2

#### Input

2018

#### Output

NOT LEAP YEAR

### C Program

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

```
int main()
```

```
{
```

```
int yr;
```

```
scanf("%d",&yr);
```

```
if(yr%100==0){
```

```
    if(yr%400==0)
```

```
    {
```

```
        printf("LEAP YEAR");
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("NOT LEAP YEAR");
```

```
    }
```

```
}
```

```
else if(yr%4==0)
```

```
{
```

```
    printf("LEAP YEAR");
```

```
}
```

```
else
```

```
{
```

```
    printf("NOT LEAP YEAR");
```

```
}
```

```
return 0;
```

```
}
```

## C++ Program

```
#include <iostream>
using namespace std;

int main()
{
    int yr;
    cin>>yr;
    if(yr%100==0){
        if(yr%400==0)
        {
            cout<<"LEAP YEAR";
        }
        else
        {
            cout<<"NOT LEAP YEAR";
        }
    }
    else if(yr%4==0)
    {
        cout<<"LEAP YEAR";
    }
    else
    {
        cout<<"NOT LEAP YEAR";
    }
    return 0;
}
```

## JAVA Program

```
import java.util.*;
import java.lang.*;
class Main {
    static boolean checkYear(int year)
    {
        if (year % 400 == 0)
            return true;
        if (year % 100 == 0)
            return false;
        if (year % 4 == 0)
            return true;
        return false;
    }
}
```

```

public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    int year;
    year=sc.nextInt();
    System.out.println( checkYear(year)? "LEAP YEAR" :
        "NOT LEAP YEAR" );
}
}

```

### Python Program

```

def checkYear(year):
    if (year % 4) == 0:
        if (year % 100) == 0:
            if (year % 400) == 0:
                return True
            else:
                return False
        else:
            return True
    else:
        return False

# Driver Code
year = int(input())
if(checkYear(year)):
    print("LEAP YEAR")
else:
    print("NOT LEAP YEAR")

```



## PROGRAM 6

There is a practice of showing a cinema in the auditorium of a college on a cinema day. A cinema day is a day where the sum of Day(D), Month(M), Year(Y) should be divisible by product of 3 and 4. On that particular day the total number of students in the hostel should be greater than 50 but less than 100. Write a C++ program to find whether the given day is cinema day or not.

### Test Case 1

#### Input

3 3 1914 76

#### Output

Cinema Day

### Test Case 2

#### Input

27 10 1995 50

#### Output

Not a Cinema Day

### C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    int d,m,y,a,sum=0;
    cin>>d>>m>>y>>a;
    sum=d+m+y;
    if(sum%12==0 && a>50 && a<100)
        cout<<"Cinema Day";
    else
        cout<<"Not a Cinema Day";
}
```

## PROGRAM 7

Write a program to find whether a given number is a strong number or not. Note: Strong number is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since,  $1! + 4! + 5! = 145$  Input Format Input consist of an Integer Output Format Output consist of String

### Test Case 1

#### Input

145

#### Output

Yes

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int num,sum=0,rem,copy,fact=1,i;
    scanf("%d",&num);
    copy=num;
    while(num!=0)
    {
        rem = num%10;
        fact=1;
        for(i=1;i<=rem;i++)
            fact=fact*i;
        sum=sum+fact;
        num=num/10;
    }
    if(copy==sum)
    {
        printf("Yes");
    }
    else
    {
        printf("No");
    }
}
```

## PROGRAM 8

Consider the following series: 1,1,2,3,4,9,8,27,16,81,32,243,64,729,128,2187...

This series is a mixture of 2 series - all the odd terms in this series forms one series and all the even terms forms yet another series. Write a program to find the Nth term in the series.

The value N in a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than value of nth term, no other character / string or message should be written to STDOUT.

### Test Case 1

#### Input

5

#### Output

4

### Test Case 2

#### Input

12

#### Output

243

## C Program

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

```
#include<math.h>
```

```
int main()
```

```
{
```

```
    int n;
```

```
    scanf("%d",&n);
```

```
    if(n % 2 == 1)
```

```
    {
```

```
        int a = 1;
```

```
        int r = 2;
```

```
        int term_in_series = (n+1)/2;
```

```
        int res = pow(2, term_in_series - 1);
```

```
        printf("%d ", res);
```

```
    }
```

```
else
```

```
{
```

```
    int a = 1;
```

```
    int r = 3;
```

```
    int term_in_series = n/2;
```

```
    int res = pow(3, term_in_series - 1);
```

```
printf("%d ", res);

}

return 0;
}
```

### **C++ Program**

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if(n % 2 == 1)
    {
        int a = 1;
        int r = 2;
        int term_in_series = (n+1)/2;
        int res = pow(2, term_in_series - 1);
        printf("%d ", res);
    }
    else
    {
        int a = 1;
        int r = 3;
        int term_in_series = n/2;
        int res = pow(3, term_in_series - 1);
        printf("%d ", res);
    }
    return 0;
}
```

## PROGRAM 9

Bob is doing a research in Pendulum. He is just pushing the pendulum aside and the pendulum started moving in to-and-fro motion. Bob will push the pendulum always towards his right side to start the oscillation. Bob wanted to calculate the distance between extreme position and the centre position of pendulum for each oscillations. He somehow calculated all the possible distance. Since he is busy in this research he is giving the task to his assistant who needs to arrange the values as instructed.

Since he is pushing the pendulum to his right always. He wanted to store that distance in the right extreme of the arrangement. And the pendulum will move towards the extreme left at that time he want that value to be stored in the left most extreme in the arrangement. And this continues till the pendulum stops. He is also sure that the distance reached at that oscillation will always be lesser than the previous oscillation towards that particular end. Write a program to arrange the distance as instructed.

### Sample Input:

5

1 3 2 5 4

### Sample Output:

4 2 1 3 5

### Explanation:

The maximum distance in the given data is 5 hence that is placed in the right most end

The next maximum element is 4 which is placed in the left most end.

Again the pendulum oscillates towards right to cover a distance of 3 and this continues.

### Test Case 1

#### Input

10 100 231 1 487 232 91 80 50 30 10

#### Output

232 100 80 30 1 10 50 91 231 487

### C++ Program

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
```

```
    int n,i,j,temp;
    cin>>n;
    int a[n],b[n];
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++)
    {
```

```

        for(j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                temp=a[j];
                a[j]=a[i];
                a[i]=temp;
            }
        }
    }
    int left=0,right=n-1,flag=1;
    for(i=n-1;i>=0;i--)
    {
        if(flag==1)
        {
            b[right--]=a[i];
            flag=0;
        }
        else
        {
            b[left++]=a[i];
            flag=1;
        }
    }
    for(i=0;i<n;i++)
        cout<<b[i]<<" ";
    return 0;
}

```

## PROGRAM 10

Given an array of numbers, find LCM of the array elements.

### Test Case 1

#### Input

4 1 2 8 3

#### Output

24

### Test Case 2

#### Input

3 1 2 3

#### Output

6

### C++ Program

```
#include<iostream>
using namespace std;
int gcd(int a, int b);
int lcm(int a[], int n);
int main()
{
    int n, arr[100], i;
    cin>>n;
    for(i=0;i<n;i++)
    {
        cin>>arr[i];
    }
    cout<<lcm(arr, n);
}
int lcm(int a[], int n)
{
    int res = 1, i;
    for (i = 0; i < n; i++)
    {
        res =res*a[i]/gcd(res,a[i]);
    }
    return res;
}
int gcd(int a, int b)
{
    if (b == 0)
    {
```

```
    return a;  
}  
return gcd(b, a%b);  
}
```



## PROGRAM 11

Write a program to verify whether a given string is a palindrome or not using recursion. A string is palindrome if the string is same as the reverse of the string. The input consists of a single line containing an string. Print yes or no based on whether the string is palindrome or not.

### Sample Input and Output:

#### Input:

21

#### Output:

no

### Test Case 1

#### Input

1122

#### Output

no

### Test Case 2

#### Input

madam

#### Output

yes

### Test Case 3

#### Input

123321

#### Output

yes

## C Program

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

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

```
int palindrome(char *str, int len, int index) {  
    if (index > len / 2) {  
        return 1;  
    }  
    if (str[index] == str[len - index - 1]) {  
        return palindrome(str, len, index + 1);  
    } else {  
        return 0;  
    }  
}
```

```

int main() {
    char str[100];
    scanf("%s", str);
    if (palindrome(str, strlen(str), 0)) {
        printf("yes\n");
    } else {
        printf("no\n");
    }
    return 0;
}

```

### C++ Program

```

#include <iostream>
#include <string.h>
using namespace std;

```

```

int palindrome(char *str, int len, int index) {
    if (index > len / 2) {
        return 1;
    }
    if (str[index] == str[len - index - 1]) {
        return palindrome(str, len, index + 1);
    } else {
        return 0;
    }
}

```

```

int main() {
    char str[100];
    cin>>str;
    if (palindrome(str, strlen(str), 0)) {
        cout<<"yes\n";
    } else {
        cout<<"no\n";
    }
    return 0;
}

```

## PROGRAM 12

Write a C++ program to find LCM of two numbers using recursion.

**Input Format:** Input consists of Integer

**Output Format:** Refer the output format

### Test Case 1

**Input**

7

5

**Output**

35

### Test Case 2

**Input**

5

5

**Output**

5

### C++ Program

```
#include<iostream>
using namespace std;
int lcm(int a, int b)
{
    if (b == 0)
        return a;
    return lcm(b, a % b);
}
int main()
{
    int a,b;
    cin>>a;
    cin>>b;
    cout<<(a*b)/lcm(a, b);
    return 0;
}
```

### PROGRAM 13

You are working in a bank. Every customer is having an account number for example 0000015697846315 but while writing it in the cheque leave it is not necessary to write the leading zeros. Write a C++ program to remove the leading zero's of the given account number. You are not allowed to use extra memory.

#### Test Case 1

##### Input

0000015697846315

##### Output

15697846315

#### Test Case 2

##### Input

00001234

##### Output

1234

#### C++ Program

```
#include <bits/stdc++.h>
#include<string>
using namespace std;
int main()
{
    int t;
    char s[20];
    cin>>s;
    int i,count=0;
    for(i=0;s[i]!='\0';i++)
    {
        if(s[i]=='0')
            count++;
        else
            break;
    }
    for(i=i;s[i]!='\0';i++)
        s[i-count]=s[i];
    s[i-count]='\0';
    cout<<s;
}
```

## PROGRAM 14

A group of people is playing a game based on vowels. If a person among them says a word or sentence others should tell the word immediately. Where while telling the word they just want to replace each vowel with the next immediate vowel lexicographical order i.e. 'a' will be replaced by 'e', 'e' will be replaced by 'i', 'i' will be replaced by 'o', 'o' will be replaced by 'u', 'u' will be replaced by 'a'. The person who says the word first with the mentioned changes is the winner of the game. Sonu is failing all the time. He is good in English but he couldn't do it properly for a lengthy sentence. Help him to win the game by writing a C++ program implementing the same.

### Test Case 1

#### Input

I am a legend

#### Output

I em e ligind

### Test Case 2

#### Input

prime

#### Output

promi

### C++ Program

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    char s[50];
    cin.getline(s,50);
    char v[5]={'a','e','i','o','u'};
    int i,j;
    for(i=0;i<s[i]!='\0';i++)
    {
        for(j=0;j<5;j++)
        {
            if(s[i]==v[j])
            {
                s[i]=v[(j+1)%5];
                break;
            }
        }
    }
    cout<<s;
}
```

## PROGRAM 15

Write a C++ program to compute net pay of an employee using structure.

**Input Format:** First input consist of name of an employee. Second input consist of Basic pay of an employee. Third input consist of Dearness Allowance of an employee. Fourth input consist of Hra of an employee.

**Note:** employee denoted as structure variable.

**Output Format:** Refer the sample output

### Test Case 1

#### Input

Jack  
90.8  
70.9  
10.9

#### Output

Net pay is : 172.6

### Test Case 2

#### Input

Raj  
67.9  
78.0  
87.9

#### Output

Net pay is : 233.8

### C++ Program

```
#include<iostream>
using namespace std;
struct employee
{
    char name[20];
    float bp;
    float da;
    float hra;
    float netpay;
}emp;
int main()
{
    employee npay(employee emp);
    employee y;
```

```
cin>>emp.name>>emp.bp>>emp.da>>emp.hra;  
y= npay(emp);  
cout<<"Net pay is : "<<y.netpay;  
}  
employee npay(employee z)  
{  
    z.netpay =z.bp + z. da +z.hra;  
    return(z);  
}
```

## PROGRAM 16

Write a C++ program to handle bank customer data by using nesting of structure.

### Input Format:

Ist input denoted as how many customer in the bank

IInd input denoted as how may account number in the Branch

IIId input consists of name of the customer

IVth input consists of address of the customer

Vth input consists of phone number of the customer

VIth input consists of bank name of the account

VIIth input consists of account\_number of the account

VIIIth input consists of balance of the account

**Output Format:** Refer the sample output

### Test Case 1

#### Input

1  
2  
Karun  
Andhra  
23  
Visa  
34000005  
75000  
canara  
680000008  
500000

#### Output

Customer Data is As:

Customer Name is: Karun

Customer Address is : Andhra

Customer Phone No is: 23

The Bank Branch Data is:

Branch Name is :Visa

Account Number is : 34000005

Balance is : 75000

Branch Name is :canara

Account Number is : 680000008

Balance is : 500000



## C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    struct customer
    {
        char name[20];
        char address[30];
        int phone_no;
        struct account
        {
            char b_name[20];
            int account_no;
            float balance;
        } acc[50];
    } cust[10];
    int i,j,n,m;
    cin>> n;
    cin>> m;
    for(i=1;i<=n;i++)
    {
        cin>> cust[i].name>> cust[i].address>>cust[i].phone_no;
        for(j=1;j<=m;j++)
        {
            cin>> cust[i].acc[j].b_name>> cust[i].acc[j].account_no >>cust[i].acc[j].balance;
        }
    }
    cout<<"Customer Data is As: ";
    for(i=1;i<=n;i++)
    {
        cout<<"\nCustomer Name is: "<< cust[i].name;
        cout<<"\nCustomer Address is : "<< cust[i].address;
        cout<<"\nCustomer Phone No is: "<< cust[i].phone_no ;
        cout<<"\nThe Bank Branch Data is:";
        for(j=1;j<=m;j++)
        {
            cout<<"\nBranch Name is : "<< cust[i].acc[j].b_name;
            cout<<"\nAccount Number is : "<< cust[i].acc[j].account_no;
            cout<<"\nBalance is : "<< cust[i].acc[j].balance;
        }
    }
}
```

## PROGRAM 17

Write a program to get a float value from the user and display it in the below-mentioned format.

**INPUT & OUTPUT FORMAT:** Input consists of 1 float value. Output must display the given input and also display the input with one, two and three decimal points.

### Test Case 1

#### Input

23.115

#### Output

23.115000

23.115

23.11

23.1

### Test Case 2

#### Input

345.67

#### Output

345.670013

345.670

345.67

345.7

### Test Case 3

#### Input

896.987

#### Output

896.987000

896.987

896.99

897.0

### Test Case 4

#### Input

3.22

#### Output

3.220000

3.220

3.22

3.2

### **C Program**

```
#include<stdio.h>
int main()
{
    float a;
    scanf("%f",&a);
    printf("%f\n",a);
    printf("%.3f\n",a);
    printf("%.2f\n",a);
    printf("%.1f\n",a);
    return 0;
}
```

### **C++ Program**

```
#include<iostream>
using namespace std;
int main()
{
    float a;
    scanf("%f",&a);
    printf("%f\n",a);
    printf("%.3f\n",a);
    printf("%.2f\n",a);
    printf("%.1f\n",a);
    return 0;
}
```

## PROGRAM 18

Selvam want to display a greeting message each day using a display on the board. He don't have computer knowledge. Any how due to size of the board constrain his greeting message will never exceed 50 characters. You want to get the greeting message and display it on the board.

### Test Case 1

#### Input

Good Morning! Have a happy day

#### Output

Good Morning! Have a happy day

### Test Case 2

#### Input

Have a pleasant day

#### Output

Have a pleasant day

### C++ Program

```
#include <iostream>
using namespace std;
```

```
int main()
{
    char a[50];
    cin.getline(a,50);
    cout<<a;
    return 0;
}
```

## PROGRAM 19

There was a large ground in center of the city which is rectangular in shape. The Corporation decides to build a Cricket stadium in the area for school and college students, But the area was used as a car parking zone. In order to protect the land from using as an unauthorized parking zone , the corporation wanted to protect the stadium by building a fence. In order to help the workers to build a fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help. Can you please help them by writing a program to find the exact length of the rope and the exact quantity of carpet that is required?

**Input format:** Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.

**Output Format:** Output Consists of two integers. The first integer corresponds to the length. The second integer corresponds to the quantity of carpet required.

### Test Case 1

#### Input

50

20

#### Output

140

1000

### Test Case 2

#### Input

70

40

#### Output

220

2800

### C Program

```
#include<stdio.h>
int main()
{
    int l,b,a,c;
    scanf("%d%d",&l,&b);
    a=(l+b)*2;
    c=l*b;
    printf("%d\n",a);
    printf("%d",c);

    return 0;
}
```

## PROGRAM 20

Training for sports day has begun and the physical education teacher has decided to conduct some team games. The teacher wants to split the students in higher secondary into equal sized teams. In some cases, there may be some students who are left out from the teams and he wanted to use the left out students to assist him in conducting the team games. For instance, if there are 50 students in a class and if the class has to be divided into 7 equal sized teams, 7 students will be there in each team and 1 student will be left out. That 1 student will assist the PET. With this idea in mind, the PET wants your help to automate this team splitting task. Can you please help him out?

**INPUT FORMAT:** Input consists of 2 integers. The first integer corresponds to the number of students in the class and the second integer corresponds to the number of teams.

**OUTPUT FORMAT:** The output consists of two integers. The first integer corresponds to the number of students in each team and the second integer corresponds to the students who are left out.

### SAMPLE INPUT:

60  
8

### SAMPLE OUTPUT:

7  
4

### Test Case 1

#### Input

24  
3

#### Output

8  
0

### Test Case 2

#### Input

54  
2

#### Output

27  
0

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
```

```
int a,b;  
cin>>a>>b;  
cout<<a/b<<"\n"<<a%b;  
  
return 0;  
}
```

## PROGRAM 21

Write a program to check whether the person is eligible to donate blood.

### Input Format:

First input consists of age

Second Input consists of weight

Third Input consists of hemoglobin level.

**Output Format:** Output consists of eligible of not

### Test Case 1

#### Input

15

44

5.4

#### Output

He or She unable to donate blood

### Test Case 2

#### Input

18

45

5.5

#### Output

He or She can donate blood

### C++ Program

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int age,weight;
```

```
    float heamo;
```

```
    scanf("%d%d%f",&age,&weight,&heamo);
```

```
    if((age>=18) && (weight>=45) && (heamo>=5.5))
```

```
        cout<<"He or She can donate blood";
```

```
    else
```

```
        cout<<"He or She unable to donate blood";
```

```
    return 0;
```

```
}
```



## PROGRAM 22

Government of India have decided to provide scholarship for Engineering students. The following are the eligibility criteria.

Students passing out on or after 2021 are eligible.

Standing arrears shall be maximum of 2.

Family Income should be less than or equal to 200000 annually

Students age should be greater than or equal to 18 and less than 21.

Score in the scholarship test should be greater than or equal to 60%.

Attendance percentage till date should be greater than or equal to 80%.

There are some relaxations provided by government for the above mentioned criteria.

If the student has more than 2 standing arrears he/she should score 80% of marks in the scholarship test and attendance should be greater than 90% till date.

If the family income is greater than 200000 but less than 250000 half the scholarship amount will be granted if he/she satisfies all the other criteria.

Write a C++ program to tell the eligibility of the student for the scholarship by providing "Eligible", "Not Eligible", "Partially Eligible"

### Input Format:

First is an Integer denoting age.

Second is an Integer denoting year of passing.

Third is an Integer denoting family income.

Fourth is an Integer input denoting standing arrears.

Fifth is an float denoting percentage of marks obtained in scholarship test.

Sixth is an float denoting attendance percentage.

**Output Format:** A string telling the eligibility as mentioned above.

### Test Case 1

#### Input

19 2022 190000 0 100 100

#### Output

Eligible

### Test Case 2

#### Input

20 2022 240000 4 85 90

#### Output

Partially Eligible

### C++ Program

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int age,year,income,arrear;
    float score,attendance;
    cin>>age>>year>>income>>arrear>>score>>attendance;
    if(age>=18 && age<21 && year>=2021 && score>=60.0 && attendance>=80.0)
    {
        if(arrear<=2 && income<=200000)
            cout<<"Eligible";
        else if(arrear>2 && income<=200000)
        {
            if(score>=80.0 && attendance>=90.0)
                cout<<"Partially Eligible";
            else
                cout<<"Not Eligible";
        }
        else if(arrear>2 && income>200000 && income<250000)
        {
            if(score>=80.0 && attendance>=90.0)
                cout<<"Partially Eligible";
            else
                cout<<"Not Eligible";
        }
        else if(arrear<=2 && income>200000 && income<250000)
            cout<<"Partially Eligible";
        else
            cout<<"Not Eligible";
    }
    else
        cout<<"Not Eligible";
    return 0;
}

```

## PROGRAM 23

Write a C++ program to find whether the given number is an Armstrong number or not. An Armstrong number is a number which is equal to the sum of all the digits raised to the power of number of digits in a given number. For example, 371 is an Armstrong number since  $3^3 + 7^3 + 1^3 = 371$ .

**INPUT & OUTPUT FORMAT:** Input consists of 1 integer. If it is an Armstrong number, display “Armstrong Number” or display “Not Armstrong Number”.

### Test Case 1

#### Input

371

#### Output

Armstrong Number

### Test Case 2

#### Input

8207

#### Output

Not Armstrong Number

## C Program

```
#include<stdio.h>
int main()
{
    int num,sum=0,rem,copy,value=1,i,digit=0;
    scanf("%d",&num);
    copy=num;
    while(num!=0)
    {
        num=num/10;
        digit++;
    }
    num=copy;
    while(num!=0)
    {
        rem = num% 10;
        value=1;
        for(i=1;i<=digit;i++)
            value=value*rem;
        sum=sum+value;
        num=num/10;
    }
    if(copy==sum)
```

```

{
    printf("Armstrong Number");
}
else
{
    printf("Not Armstrong Number");
}
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main()
{
    int num,sum=0,rem,copy,value=1,i,digit=0;
    scanf("%d",&num);
    copy=num;
    while(num!=0)
    {
        num=num/10;
        digit++;
    }
    num=copy;
    while(num!=0)
    {
        rem = num% 10;
        value=1;
        for(i=1;i<=digit;i++)
            value=value*rem;
        sum=sum+value;
        num=num/10;
    }
    if(copy==sum)
    {
        printf("Armstrong Number");
    }
    else
    {
        printf("Not Armstrong Number");
    }
}

```

## PROGRAM 24

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

**Input Format:** First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

**Output Format:** Print "Yes" in the case Alice wins, else print "No".

### Constraints:

$1 \leq T \leq 1000$

$1 \leq N \leq 10000$

### Test Case 1

#### Input

3

1

6

7

#### Output

Yes

Yes

No

### Test Case 2

#### Input

5

9

2

7

4

5

#### Output

Yes

No

No

Yes

No

### Test Case 3

#### Input

4  
997  
998  
999  
1000

#### Output

No  
Yes  
No  
No

### C++ Program

```
#include<iostream>
using namespace std;
```

```
int solve(int N)
{
    int x,y;
    int a;
    int k;
    x=N/4;
    y=N%4;
    if(y==0)
        return x%2;
    else
    {
        a=0;
        if(x%2)
        {
            for(k=1;k<y;++k)
                a++k;
            if(a==y)
                return 1;
            else
                return 0;
        }
        else
        {
            for(k=0;k<y;++k)
                a++k;
            if(a==y)
```

```

        return 1;
    else
        return 0;
    }
}
}

int main()
{
    int T;
    int N;
    int result;
    int i;
    cin>>T;
    if(T<1||T>1000)
        return 0;
    for(i=0;i<T;i++)
    {
        cin>>N;

        if(N<1||N>10000)
            return 0;
        result=solve(N);
        if(result)
            cout<<"Yes"<<endl;
        else
            cout<<"No"<<endl;
    }
    return 0;
}

```

## PROGRAM 25

A famous company is going to select a new Managing Director. All the eligible candidates are waiting in the meeting room. Everyone is seated in a random order linearly. Everyone are cunning and want's to become the Managing Director. If the founder of that company tell's a person name directly definitely that is going to become a serious problem. But the founder already have a person on his mind. But he was thinking for a creative idea to convey the same. Suddenly he notices a pattern follows and he find a way to tell who is the new MD.

He say's "the next MD of this company is the person who haves the age greater than all the other person to his right side."

Write the program to find the new MD of that company where you are provided with the n persons age in order similar to their seating arrangement. In case if the founder couldn't find the next MD print, "The promotion stands cancelled".

**Note:** There will be at least 2 person waiting in the meeting room.

### Sample Input:

7  
56 53 52 58 51 56 57

### Sample Output:

4

### Explanation:

In the given input person sitting 4th from the left is having the age as 58. Where all the others to his right were less than his age(51,56,57).

### Test Case 1

#### Input

5 1 2 3 4 5

#### Output

The promotion stands cancelled

### C Program

```
#include<stdio.h>
int main()
{
    int n,i,j,flag=0,count=0;
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n-1;i++)
    {
        flag=0;
        for(j=i+1;j<n;j++)
```



```

    {
        if(a[i]<a[j])
        {
            flag=1;
            break;
        }
    }
    if(flag==0)
    {
        printf("%d",i+1);
        count++;
        break;
    }
}
if(count==0)
    printf("The promotion stands cancelled");
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main()
{
    int n,i,j,flag=0,count=0;
    cin>>n;
    int a[n];
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n-1;i++)
    {
        flag=0;
        for(j=i+1;j<n;j++)
        {
            if(a[i]<a[j])
            {
                flag=1;
                break;
            }
        }
    }
    if(flag==0)
    {
        cout<<i+1;
        count++;
    }
}

```

```
        break;
    }
}
if(count==0)
    cout<<"The promotion stands cancelled";
}
```

## PROGRAM 26

There is a treasure full of gold and diamonds. Jack Sparrow successfully reached the cave with his black pearl. He is reading a hint on a box that he need to tilt it 90 degree clock wise. But he don't know what to tilt. Finally when he reaches the door there is a dial pad. He remember the hint in the box and he opened the box where it has numbers printed in N x N form. Now he is tilting the same to 90 degree and seeing the numbers. He couldn't see and type on same time. So write a C++ program to print the 2D matrix which will be helpful for him to type easily.

### Test Case 1

#### Input

3 1 2 3 4 5 6 7 8 9

#### Output

7 4 1 8 5 2 9 6 3

### Test Case 2

#### Input

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

#### Output

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

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int arr[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            cin>>arr[i][j];
    }
    for (int i=0;i<n/2;i++)
    {
        for (int j=i;j<n-i-1;j++)
        {
            int temp=arr[i][j];
            arr[i][j]=arr[n-1-j][i];
            arr[n-1-j][i]=arr[n-1-i][n-1-j];
            arr[n-1-i][n-1-j]=arr[j][n-1-i];
            arr[j][n-1-i]=temp;
        }
    }
}
```

```
for(int i=0;i<n;i++)  
{  
    for(int j=0;j<n;j++)  
        cout<<arr[i][j]<<" ";  
}  
return 0;  
}
```

## PROGRAM 27

For enhancing the book reading, school distributed story books to students as part of the Children's day celebrations. To increase the reading habit, the class teacher decided to exchange the books every weeks so that everyone will have a different book to read. She wants to know how many possible exchanges are possible.

If they have 4 books and students, the possible exchanges are 9.  $B_i$  is the book of  $i$ -th student and after the exchange, he should get a different book, other than  $B_i$ .

B1 B2 B3 B4 – first state, before exchange of the books

B2 B1 B4 B3

B2 B3 B4 B1

B2 B4 B1 B3

B3 B1 B4 B2

B3 B4 B1 B2

B3 B4 B2 B1

B4 B1 B2 B3

B4 B3 B1 B2

B4 B3 B2 B1

Find the number of possible exchanges, if the books are exchanged so that every student will receive a different book.

### Constraints

$1 \leq N \leq 1000000$

**Input Format:** Input contains one line with N, indicates the number of books and number of students.

**Output Format:** Output the answer modulo 100000007. Refer the sample output for formatting

### Test Case 1

#### Input

4

#### Output

9

### Test Case 2

#### Input

10

#### Output

1334961

### C++ Program

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
long long int countDer(long long int n)
```

```

{
// Base cases
if (n == 1)
    return 0;
if (n == 0)
    return 1;
if (n == 2)
    return 1;

// countDer(n) = (n-1)[countDer(n-1) + der(n-2)]
return ((n - 1) * (countDer(n - 1) +
    countDer(n - 2)))% 1000000007;
}

int main()
{
    long long int n;
    cin >> n;
    cout <<countDer(n);
    return 0;
}

```

## PROGRAM 28

Imagine the field is a 2D plane. Each cell is either water 'W' or a tree 'T'. A forest is a collection of connected trees. Two trees are connected if they share a side i.e. if they are adjacent to each other. Your task is, given the information about the field, print the size of the largest forest. Size of a forest is the number of trees in it. See the sample case for clarity.

**INPUT:** First line contains the size of the matrix N. The next N lines contain N characters each, either 'W' or 'T'.

**OUTPUT:** Print the size of the biggest forest.

### CONSTRAINTS:

$1 \leq N \leq 1000$

### Sample Input:

```
5
TTTWW
TWWTT
TWWTT
TWTTT
WWTTT
```

### Sample Output:

```
10
```

**Explanation:** The forest on the top left has 6 trees but the forest on the bottom right is bigger with 10 trees.

### Test Case 1

#### Input

```
6 TTTWWT TWWTTT TWWTTT TWTTTW WWTTTW TWWTWW
```

#### Output

```
14
```

### C++ Program

```
#include<bits/stdc++.h>
using namespace std;
int count=0;
int N;
char mtr[1000][1000];
void count_adj(int i,int j)
{
    if(i<0 || j<0 || i>N || j>N)
        return ;
    if(mtr[i][j]=='T')
    {
```

```

        mtr[i][j]='W';
        ::count++;
        count_adj(i-1,j);
        count_adj(i+1,j);
        count_adj(i,j+1);
        count_adj(i,j-1);
    }
}
int main()
{
    cin>>N;
    for(int i=0; i<N; i++)
        cin>>mtr[i];
    int max_count=0;
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<N; j++)
        {
            if(mtr[i][j]=='T')
            {
                count_adj(i,j);
                if(max_count< ::count)
                    max_count= ::count;
                ::count=0;
            }
        }
    }
    cout<<max_count;
    return 0;
}

```



## PROGRAM 29

Robert is expert in strings where he challenges everyone to write a program for the below implementation.

Two strings and comprising of lower case English letters are compatible if they are equal or can be made equal by following this step any number of times:

Select a prefix from the string (possibly empty), and increase the alphabetical value of all the characters in the prefix by the same valid amount.

For example if the string is abc and we select the prefix ab then we can convert it to bcc by increasing the alphabetical value by 1. But if we select the prefix abc then we cannot increase the alphabetical value.

Your task is to determine if given strings are compatible.

### Input format

First line: String A

Next line: String B

**Output format:** For each test case, print YES if string can be converted to string , otherwise print NO.

**Constrain:**  $1 \leq (\text{len of A,B}) < 1000005$

### Sample Input:

abaca

cdbda

### Sample Output:

YES

### Explanation:

The string abaca can be converted to bcbda in one move and to cdbda in the next move.

### Test Case 1

#### Input

bcd ef cdefg

#### Output

YES

### C++ Program

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char str1[1000005];
    char str2[1000005];
    cin>>str1>>str2;
    int max = str2[0] - str1[0];
```

```
if(max > 0)
{
    for(int i =1;i<strlen(str1);i++)
    {
        int x = str2[i] - str1[i];
        if(x > max)
        {
            cout<<"NO";
            return 0;
        }
    }
    cout<<"YES";
}
else
    cout<<"NO";
return 0;
}
```

### PROGRAM 30

Write a C++ program to rotate the string in the specified direction and print the output.

#### Input Format:

First line contains the String A.

Second line contains the number of positions you have to shift the elements in the string.

Third line contains the direction either 'L' or 'R'.

**Output Format:** Print the rotated string

**Constrain:**  $1 \leq \text{len}(A) < 1000$

#### Sample Input:

movies 3 L

#### Sample Output:

iesmov

#### Test Case 1

##### Input

welcome 5 L

##### Output

mewelco

#### C++ Program

```
#include<iostream>
#include<string.h>
using namespace std;
void leftrotation(char *a, char *b, int lr)
{
    int len = strlen(a),i;
    for(i=0;i<len;i++)
        b[(len-lr+i)%len]=a[i];
    b[len]='\0';
}
void rightrotation(char *a, char *b, int rr)
{
    int len = strlen(a),i;
    for(i=0;i<len;i++)
        b[(i+rr)%len]=a[i];
    b[len]='\0';
}
int main()
{
```

```
char a[1000],b[1000];
cin>>a;
int n;
cin>>n;
char d;
cin>>d;
if(d=='L')
{
    leftrotation(a,b,n);
}
else if(d=='R')
{
    rightrotation(a,b,n);
}
cout<<b;
return 0;
}
```

### PROGRAM 31

Write a C++ program to calculate difference between two time period using structure .

#### Input Format:

First input consist of hours of time1

Second input consist of minutes of time1

Third input consist of seconds of time1

Fourth input consist of hours of time2

Fifth input consist of minutes of time2

Sixth input consist of seconds of time2

**Note:** hours ,minutes and seconds denoted as structure variable.

**Output Format:** Refer sample output format

#### Test Case 1

##### Input

12

45

56

2

39

45

##### Output

TIME DIFFERENCE: 12:45:56 - 2:39:45 = 10:6:11

#### Test Case 2

##### Input

4

56

60

5

4

10

##### Output

TIME DIFFERENCE: 4:56:60 - 5:4:10 = -1:52:50

#### C++ Program

```
#include <iostream>
```

```
using namespace std;
```

```
struct TIME
```

```
{
```

```
    int seconds;
```

```

int minutes;
int hours;
};
void computeTimeDifference(struct TIME, struct TIME, struct TIME *);
int main()
{
    struct TIME t1, t2, difference;
    cin >> t1.hours >> t1.minutes >> t1.seconds;
    cin >> t2.hours >> t2.minutes >> t2.seconds;
    computeTimeDifference(t1, t2, &difference);
    cout << "TIME DIFFERENCE: " << t1.hours << ":" << t1.minutes << ":" << t1.seconds;
    cout << " - " << t2.hours << ":" << t2.minutes << ":" << t2.seconds;
    cout << " = " << difference.hours << ":" << difference.minutes << ":" << difference.seconds;
    return 0;
}
void computeTimeDifference(struct TIME t1, struct TIME t2, struct TIME *difference)
{
    if(t2.seconds > t1.seconds)
    {
        --t1.minutes;
    }
    difference->seconds = t1.seconds - t2.seconds;
    if(t2.minutes > t1.minutes)
    {
        --t1.hours;
    }
    difference->minutes = t1.minutes-t2.minutes;
    difference->hours = t1.hours-t2.hours;
}

```

## PROGRAM 32

Write a C++ program to read and display student data using union.

### Input Format:

First input consist of grade of student

Second input consist of roll number of student

Third input consist of mark of student

Fourth input consist of fees of student

**Output Format:** Displays student data with input values

### Test Case 1

#### Input

A

15EC110

78.98

25000

#### Output

Grade is :A

Rollno is:15EC110

Marks are :78.98

Fees paid is: 25000

### Test Case 2

#### Input

C

15EC089

56

50000

#### Output

Grade is :C

Rollno is:15EC089

Marks are :56

Fees paid is: 50000

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    union student
    {
        char grade;
```

```
    char rollno[20];  
    float marks;  
    double fees;  
}s;  
cin>>s.grade;  
cout<<"Grade is : "<<s.grade<<endl;  
cin>>s.rollno;  
cout<<"Rollno is: "<<s.rollno<<endl;  
cin>>s.marks;  
cout<<"Marks are : "<<s.marks<<endl;  
cin>>s.fees;  
cout<<"Fees paid is: " <<s.fees<<endl;  
}
```



### PROGRAM 33

In this lockdown a family of N members decided to play a game the rules of which are :-

All N members are made to sit uniformly in a circle (ie. from 1 to N in clockwise direction).

The game start with the person sitting at first position.

A song is played in the background. The lyrics of the song are denoted by a string which consists of only letters 'x' and 'y'. Assume that each lyric of the song is a single letter.

If the lyric 'x' occurs in the song, the member who is currently holding the Parcel passes it on to the next member. This passing takes place in clockwise direction.

If the lyric 'y' occurs in the song, the member who is currently holding the Parcel loses his/her chances of winning the game. He/she hands over the parcel to the next member (in clockwise direction) and moves out.

The game continues until a single member survives in the end. He/she will be the winner of the game.

Note that the song repeats continuously ie. while the game is going on, if at all the song ends, the stereo system will automatically start playing the song from the start without any delay.

You have to find out the member who wins the game.

**Input:** The input consists of 2 lines. The first line consists of N, the member of family in the class. The next line consists of a string denoting the lyrics of the song the teacher plays.

**Output:** Print a single integer denoting the roll number of the student who wins the game.

**Constraints:**  $2 \leq N \leq 100000$

$1 \leq |S| \leq 10000$ , where  $|S|$  denotes the length of the input string. It is guaranteed that at least 1 lyric in the song will be a 'y'

#### Sample Input:

3

xyx

#### Sample Output:

1

#### Explanation:

Starting from 1 lyrics : 'x' therefore he passes the ball to 2nd

2nd turn lyrics : 'y' therefore 2nd member gets out of game and passes to 3rd

3rd turn lyrics : 'x' therefore 3rd passes ball to first.

4th turn lyrics : 'x' passes to 3rd

5th turn lyrics: 'y' therefore gets eliminated.

Hence person sitting at position 1 won this game.

#### Test Case 1

##### Input

6 xxyxyx

##### Output

2

## C++ Program

```
#include<bits/stdc++.h>
#include<string>
using namespace std;
#define ll long long
ll n,slen;

void fun (ll size,ll k,string s, ll l, ll *ingame)
{
    if(size==1)
    {
        ll i=1;
        while(ingame[i]!=1)
        {i++;

        }
        cout<<i<<endl;
        return;
    }
    if(l==slen) l=0;
    if(k==n+1) k=1;
    if(ingame[k]==0)
    {

        fun(size,k+1,s,l,ingame);
    }
    else
    {
        if(s[l]=='x')
        fun(size,k+1,s,l+1,ingame);
        else{
            ingame[k]=0;
            fun(size-1,k+1,s,l+1,ingame);
        }
    }
    return;
}

int main()
{
    cin>>n;
    string s;
    cin>>s;
    slen=s.length();
```

```
ll ingame[n+1];  
for(int i =1;i<=n;i++)  
{  
    ingame[i]=1;  
}  
fun(n,1,s,0,ingame);  
}
```

### PROGRAM 34

Write a program to check whether the given number is prime or not, If number is prime, find the square root of the number else print given number is not a prime number.

**Input Format:** Input consists of integer

**Output Format:** Refer the sample output format

**Sample Input:**

8

**Sample Output:**

8 is not a prime number

**Test Case 1**

**Input**

7

**Output**

2.65

### C Program

```
#include <stdio.h>
#include<math.h>
int main()
{
    int n, i, flag = 0;
    double result;
    scanf("%d",&n);
    for(i = 2; i <= n/2; ++i)
    {
        // condition for nonprime number
        if(n%i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0){
        result = sqrt(n);
        printf("%.2lf",result);
    }
    else
        printf("%d is not a prime number",n);
    return 0;
}
```

## C++ Program

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
    int n, i, flag = 0;
    double result;
    cin>>n;
    for(i = 2; i <= n/2; ++i)
    {
        if(n%i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0){
        result = sqrt(n);
        printf("%.2lf",result);
    }
    else
        cout<<n<<" is not a prime number";
    return 0;
}
```

### PROGRAM 35

Write a program to find the sum of factorial of each digit in the given number.

**Input Format:** Input consists of integer

**Output Format:** Refer the sample output format

#### Test Case 1

**Input**

56

**Output**

840

#### C Program

```
#include <stdio.h>
int main()
{
    int num, a[10], i=0, j=0, rem, c, fact=1, p, sum, len=0;
    scanf("%d", &num);
    while (num > 0)
    {
        rem = num % 10;
        a[len] = rem;
        num /= 10;
        len ++;
    }
    for(j = 0; j < len; j ++){
        for (c = 1; c <= a[j]; c ++){
            fact = fact * c;
        }
        a[j] = fact;
        fact = 1;
    }
    sum = 0;
    for(j = 0; j < len; j ++){
        sum = sum + a[j];
    }
    printf("%d", sum);
}
```

## C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    int num, a[10], i=0, j=0, rem, c, fact=1, p, sum, len=0;
    cin>>num;
    while (num > 0)
    {
        rem = num % 10;
        a[len] = rem;
        num /= 10;
        len ++;
    }
    for(j = 0;j < len;j ++){
        for (c = 1; c <= a[j]; c ++){
            fact = fact * c;
        }
        a[j] = fact;
        fact = 1;
    }
    sum = 0;
    for(j = 0;j < len;j ++){
        sum = sum + a[j];
    }
    cout<<sum;
}
```

## PROGRAM 36

Find the area of a triangle and check whether the area is palindrome or not.

### Test Case 1

#### Input

4 6 8

#### Output

Area of a triangle = 11

Area is palindrome

### Test Case 2

#### Input

11 11 11

#### Output

Area of a triangle = 44

Area is palindrome

### C Program

```
#include <stdio.h>
#include <math.h>
int main()
{
    int s, a, b, c, area,num,d,pal=0;
    scanf("%d %d %d", &a, &b, &c);
    /* compute s */
    s = (a + b + c) / 2;
    area = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("Area of a triangle = %d ", area);
    num=area;
    /*reverse the area*/
    while(area!=0)
    {
        d=area%10;
        pal=d+(pal*10);
        area=area/10;
    }
    if(num==pal)
    {
        printf("\nArea is palindrome ");
    }
    else
    {
        printf("\nArea is not palindrome ");
    }
}
```



```

    }
    return 0;
}

```

### C++ Program

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int s, a, b, c, area,num,d,pal=0;
    cin>>a>>b>>c;
    s = (a + b + c) / 2;
    area = sqrt(s * (s - a) * (s - b) * (s - c));
    cout<<"Area of a triangle = "<< area<<" ";
    num=area;
    while(area!=0)
    {
        d=area%10;
        pal=d+(pal*10);
        area=area/10;
    }
    if(num==pal)
    {
        cout<<"\nArea is palindrome ";
    }
    else
    {
        cout<<"\nArea is not palindrome ";
    }
    return 0;
}

```

## PROGRAM 37

Write a program to find whether the given string is the anagram of the first string.

### Test Case 1

#### Input

eat ate

#### Output

Anagram

### Test Case 2

#### Input

eat abc

#### Output

Not Anagram

### C++ Program

```
#include<iostream>
using namespace std;
int find_anagram(char [], char []);
int main()
{
    char array1[100], array2[100];
    int flag;
    cin>>array1;
    cin>>array2;
    flag = find_anagram(array1, array2);
    if (flag == 1)
        cout<<"Anagram";
    else
        cout<<"Not Anagram";
    return 0;
}
int find_anagram(char array1[], char array2[])
{
    int num1[26] = {0}, num2[26] = {0}, i = 0;
    while (array1[i] != '\0')
    {
        num1[array1[i] - 'a']++;
        i++;
    }
    i = 0;
    while (array2[i] != '\0')
    {
```

```
    num2[array2[i] - 'a']++;  
    i++;  
}  
for (i = 0; i < 26; i++)  
{  
    if (num1[i] != num2[i])  
        return 0;  
}  
return 1;  
}
```

## PROGRAM 38

Write a program to calculate the square root of a number without using sqrt.h().

### Test Case 1

#### Input

16

#### Output

4.00

### C Program

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n;
    scanf("%d", &n);
    float i=0.00;
    while(i*i<=n)
    {
        i=i+0.001;
    }
    i=i-0.001;
    printf("%.2f",i);
    return 0;
}
```

### C++ Program

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

```
using namespace std;
```

```
int main()
{
    int n;
    cin>>n;
    float i=0.00;
    while(i*i<=n)
    {
        i=i+0.001;
    }
    i=i-0.001;
```

```
printf("%.2f",i);
return 0;
}
```

## **JAVA Program**

```
import java.util.*;
class Main
{

static double Square(double n,
                    double i, double j)
{
    double mid = (i + j) / 2;
    double mul = mid * mid;

    if ((mul == n) || (Math.abs(mul - n) < 0.00001))
        return mid;

    else if (mul < n)
        return Square(n, mid, j);

    else
        return Square(n, i, mid);
}

static void findSqrt(double n)
{
    double i = 1;

    boolean found = false;
    while (!found)
    {
        if (i * i == n)
        {
            System.out.println(i);
            found = true;
        }

        else if (i * i > n)
        {
            double res = Square(n, i - 1, i);
            System.out.printf("%.2f", res);
            found = true;
        }
    }
}
```

```

    }
    i++;
}
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    double n=sc.nextDouble();
    findSqrt(n);
}
}

```

## Python

```

import math
def Square(n, i, j):

    mid = (i + j) / 2;
    mul = mid * mid;

    if ((mul == n) or (abs(mul - n) < 0.00001)):
        return mid;

    elif (mul < n):
        return Square(n, mid, j);

    else:
        return Square(n, i, mid);

def findSqrt(n):
    i = 1;

    found = False;
    while (found == False):

        if (i * i == n):
            print(i);
            found = True;

        elif (i * i > n):

            res = Square(n, i - 1, i);
            print ("{0:.2f}".format(res))

```

```
        found = True
    i += 1;

n = int(input());
findSqrt(n);
```

## PROGRAM 39

Write a program to convert the given binary number to its equivalent octal value.

**Input Format:** Input consists of integer

**Output Format:** Refer the sample output format

### Test Case 1

#### Input

10101

#### Output

25

### C Program

```
#include<stdio.h>
int power_fn(int base,int exp)
{
    int i,sum=1;
    for(i=0;i<exp;i++)
    {
        sum=sum*base;
    }
    return sum;
}
int main()
{
    int binary,sum=0,rem,exp=0;
    scanf("%d",&binary);
    while(binary!=0)
    {
        rem=binary%10;
        sum=sum+rem*power_fn(2,exp);
        exp++;
        binary=binary/10;
    }
    int array[100],i=0,j;
    while(sum!=0)
    {
        array[i++]=sum%8;
        sum=sum/8;
    }
    for(j=i-1;j>=0;j--)
        printf("%d",array[j]);
}
```



## C++ Program

```
#include<iostream>
#include<stdio.h>
using namespace std;
int power_fn(int base,int exp)
{
    int i,sum=1;
    for(i=0;i<exp;i++)
    {
        sum=sum*base;
    }
    return sum;
}
int main()
{
    int binary,sum=0,rem,exp=0;
    cin>>binary;
    while(binary!=0)
    {
        rem=binary%10;
        sum=sum+rem*power_fn(2,exp);
        exp++;
        binary=binary/10;
    }
    int array[100],i=0,j;
    while(sum!=0)
    {
        array[i++]=sum%8;
        sum=sum/8;
    }
    for(j=i-1;j>=0;j--)
    cout<<array[j];
}
```

## JAVA Program

```
import java.util.*;
public class Main
{
    public static int binary_to_octal( int binary)
    {
        int octal = 0, decimal = 0, i = 0;

        while(binary != 0)
```

```

{
decimal += (binary%10) * Math.pow(2,i);
++i;
binary/=10;
}

i = 1;

while (decimal != 0)
{
octal += (decimal % 8) * i;
decimal /= 8;
i *= 10;
}

return octal;
}

public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
int binary = sc.nextInt();
System.out.print( binary_to_octal(binary));
}
}

```

## Python

```

import math
def binary_to_octal(binary):
    octal = 0
    decimal = 0
    i = 0
    while(binary != 0):
        decimal += (binary%10) * math.pow(2,i);
        i = i + 1
        binary = int(binary/10)

    i = 1

    while (decimal != 0):
        octal += (decimal % 8) * i;
        decimal = int(decimal/8)
        i *= 10;

```

```
    return octal;

binary = int(input())
print(int(binary_to_octal(binary)))
```

## PROGRAM 40

Given a string S and a character C, write a program to count the number of sub-strings of S that contains only the character C.

### Input:

First line contains the string S

Second line contains the character C.

**Output:** Single Integer denoting the no.of sub-string containing only the given character C.

### Input:

0110111

1

### Output:

9

### Explanation:

The sub-strings containing only '1' are:

“1” — 5 times

“11” — 3 times

“111” — 1 time

Hence, the count is 9.

### Test Case 1

#### Input

programming r

#### Output

2

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
void countSubString(string S, char C)
{
    int count = 0;
    int conCount = 0;
    for (int i = 0; S[i] != '\0'; i++)
    {
        char ch = S[i];
        if (ch == C)
            conCount++;
        else
        {
            count += conCount;
            conCount = 0;
        }
    }
    count += conCount;
}
```

```

        count += (conCount * (conCount + 1))/ 2;
        conCount = 0;
    }
}
count += (conCount* (conCount + 1))/ 2;
cout << count;
}
int main()
{
    string S;
    char C;
    getline(cin,S);
    cin>>C;
    countSubString(S, C);
    return 0;
}

```

## PROGRAM 41

Program to Count words and characters. Get a string from the user and print number of words and characters.

### Test Case 1

#### Input

Youtube Channel

#### Output

14

2

### C Program

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

### C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    char str[100];
    int i = 0, l = 0, f = 1;
    scanf("%[^\\n]s",str);
    for(i = 0; str[i] != '\\0'; i++)
    {
```

```
    if(str[i]!=' ')
        l = l + 1;
}
printf("%d\n", l);
for(i = 0; i <= l-1; i++)
{
    if(str[i] == ' ')
        f = f + 1;
}
printf("%d", f);
return 0;
}
```

## PROGRAM 42

Write a program to display the characters in prime position of a given string.

**Input Format:** Input consists of Integer

**Output Format:** Refer the sample output format

### Sample Input:

Computer

### Sample Output :

o m u e

### Test Case 1

#### Input

Characters in Prime Position

#### Output

h a a t n i e s

### C Program

```
#include <stdio.h>
#include <string.h>
int main()
{
    int i, j, len, count = 0;
    char str[50];
    scanf("%[^\n]s", str);
    len = strlen(str);
    for (i = 2; i <= len; i++)
    {
        count = 0;
        for (j = 2; j <= len; j++)
        {
            if (i % j == 0)
            {
                count++;
            }
        }
        if (count == 1)
        {
            printf("%c ", str[i-1]);
        }
    }
}
```



## C++ Program

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int i, j, len, count = 0;
    char str[50];
    scanf("%[^\\n]s", str);
    len = strlen(str);
    for (i = 2; i <= len; i++)
    {
        count = 0;
        for (j = 2; j <= len; j++)
        {
            if (i % j == 0)
            {
                count++;
            }
        }
        if (count == 1)
        {
            cout<<str[i-1]<<" ";
        }
    }
}
```

### PROGRAM 43

Given an array `arr[]` of  $N$  integers arranged in a circular fashion. Your task is to find the maximum contiguous subarray sum.

**Input:** First line of input contains a single integer  $T$  which denotes the number of test cases. First line of each test case contains a single integer  $N$  which denotes the total number of elements. Second line of each test case contains  $N$  space separated integers denoting the elements of the array.

**Output:** For each test case print the maximum sum obtained by adding the consecutive elements.

#### Constraints:

$1 \leq T \leq 101$

$1 \leq N \leq 106$

$-106 \leq \text{Arr}[i] \leq 106$

**Testcase 1:** Starting from the last element of the array, i.e, 12, and moving in circular fashion, we have max subarray as 12, 8, -8, 9, -9, 10, which gives a maximum sum as 22.

#### Test Case 1

##### Input

3 7 8 -8 9 -9 10 -11 12 8 10 -3 -4 7 6 5 -4 -1 8 -1 40 -14 7 6 5 -4 -1

##### Output

22

23

52

#### Test Case 2

##### Input

1 5 10 20 -10 -20 30

##### Output

60

#### C++ Program

```
#include<bits/stdc++.h>
using namespace std;
int max_sum(int a[],int n)
{
    int sum[n];
    sum[0] = a[0];
    for(int i=1;i<n;i++)
    {
        sum[i] = sum[i-1]+a[i];
        // cout<<sum[i]<<" ";
    }
}
```

```

    }
    int mi = 0;
    int ma = -200000;
    int cur = -200000;
    //cout<<endl;
    for(int i=0;i<n;i++)
    {
        if(sum[i] < mi )
            mi = sum[i];
        if(sum[i] - mi > cur)
        {
            ma = sum[i];
            cur = sum[i] - mi;
        }
        //cout<<mi<<" "<<ma<<" "<<cur<<endl;
    }
    return cur;
}

int main() {
    //code
    int t;
    cin>>t;
    while(t-->0)
    {
        int n;
        cin>>n;
        int a[n];
        bool ch = 0;
        for(int i=0;i<n;i++)
        {
            cin>>a[i];
            if(a[i] >= 0)
                ch = 1;
        }
        if(ch == 0)
        {
            cout<<*max_element(a,a+n)<<endl;
            continue;
        }

        int x = 0;
        x = max_sum(a,n);
        int sum = a[0];
        a[0] = -a[0];
    }
}

```

```
for(int i=1;i<n;i++)
{
    sum = sum+a[i];
    a[i] = -1*a[i];
}
sum = sum+max_sum(a,n);
cout<<max(x,sum)<<endl;
}
return 0;
}
```

## PROGRAM 44

Write a program to print the stars in a diamond fashion.

### Test Case 1

#### Input

4

#### Output

```
  *
 ***
*****
*****
 ***
  *
```

### C Program

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

```
int main()
```

```
{
```

```
    int i, j, rows;
```

```
    scanf("%d", &rows);
```

```
    /* Iterate through rows */
```

```
    for(i=1; i<=rows; i++)
```

```
    {
```

```
        /* Print leading spaces */
```

```
        for(j=i; j<rows; j++)
```

```
        {
```

```
            printf(" ");
```

```
        }
```

```
        /* Print star */
```

```
        for(j=1; j<=(2*i-1); j++)
```

```
        {
```

```
            printf("*");
```

```
        }
```

```
        /* Move to next line */
```

```
        printf("\n");
```

```
    }
```

```
    for(i=rows-1; i>=1; i--)
```

```

{
    /* Print leading spaces */
    for(j=i; j<rows; j++)
    {
        printf(" ");
    }

    /* Print star */
    for(j=1; j<=(2*i-1); j++)
    {
        printf("*");
    }

    /* Move to next line */
    printf("\n");
}
return 0;
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main(){
    int i, j, rows;

    cin>>rows;

    /* Iterate through rows */
    for(i=1; i<=rows; i++)
    {
        /* Print leading spaces */
        for(j=i; j<rows; j++)
        {
            cout<<" ";
        }

        /* Print star */
        for(j=1; j<=(2*i-1); j++)
        {
            cout<<"*";
        }

        /* Move to next line */

```

```

        cout<<"\n";
    }
    for(i=rows-1; i>=1; i--)
    {
        /* Print leading spaces */
        for(j=i; j<rows; j++)
        {
            cout<<" ";
        }

        /* Print star */
        for(j=1; j<=(2*i-1); j++)
        {
            cout<<"*";
        }

        /* Move to next line */
        cout<<"\n";
    }
    return 0;
}

```

## JAVA Program

```

import java.util.Scanner;
class Main{
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        int i, j;
        int rows = in.nextInt();
        /* Iterate through rows */
        for(i=1; i<=rows; i++)
        {
            /* Print leading spaces */
            for(j=i; j<rows; j++)
            {
                System.out.print(" ");
            }
            /* Print star */
            for(j=1; j<=(2*i-1); j++)
            {
                System.out.print("*");
            }
            /* Move to next line */

```

```

        System.out.println();
    }
    for(i=rows-1; i>=1; i--)
    {
        /* Print leading spaces */
        for(j=i; j<rows; j++)
        {
            System.out.print(" ");
        }

        /* Print star */
        for(j=1; j<=(2*i-1); j++)
        {
            System.out.print("*");
        }

        /* Move to next line */
        System.out.println();
    }
}
}

```



## PROGRAM 45

Given a string, the task is to find whether it contains an additive sequence or not. A string contains an additive sequence if its digits can make a sequence of numbers in which every number is addition of previous two numbers. A valid string should contain at least three digit to make one additive sequence.

### Sample Input:

235813

### Sample Output:

true

### Explanation:

$2 + 3 = 5$ ,  $3 + 5 = 8$ ,  $5 + 8 = 13$

### Test Case 1

#### Input

12345678

#### Output

false

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
bool isValid(string num)
{
    if (num.size() > 1 && num[0] == '0')
        return false;
    return true;
}
int val(string a, int pos)
{
    if (pos >= a.length())
        return 0;
    return (a[pos] - '0');
}
string addString(string a, string b)
{
    string sum = "";
    int i = a.length() - 1;
    int j = b.length() - 1;
    int carry = 0;
    while (i >= 0 || j >= 0)
    {
        int t = val(a, i) + val(b, j) + carry;
```

```

        sum += (t % 10 + '0');
        carry = t / 10;
        i--; j--;
    }
    if (carry)
        sum += (carry + '0');
    reverse(sum.begin(), sum.end());
    return sum;
}

bool checkAddition(list<string>& res, string a, string b, string c)
{
    if (!isValid(a) || !isValid(b))
        return false;
    string sum = addString(a, b);
    if (sum == c)
    {
        res.push_back(sum);
        return true;
    }

    if (c.size() <= sum.size() ||
        sum != c.substr(0, sum.size()))
        return false;
    else
    {
        res.push_back(sum);
        return checkAddition(res, b, sum, c.substr(sum.size()));
    }
}

list<string> additiveSequence(string num)
{
    list<string> res;
    int l = num.length();
    for (int i = 1; i <= l/2; i++)
    {
        for (int j = 1; j <= (l - i)/2; j++)
        {
            if (checkAddition(res, num.substr(0, i), num.substr(i, j), num.substr(i + j)))
            {
                res.push_front(num.substr(i, j));
                res.push_front(num.substr(0, i));
                return res;
            }
        }
    }
}

```

```
    }  
    res.clear();  
    return res;  
}  
int main()  
{  
    string num;  
    getline(cin,num);  
    list<string> res = additiveSequence(num);  
    if(res.empty())  
        cout<<"false";  
    else  
        cout<<"true";  
}
```

## PROGRAM 46

Given an array of integers, sort the first half of the array in ascending order and second half in descending order.

### Sample Input:

Size of an array: 6

Arr[] = { 3, 5, 6, 2, 1, 4}

### Sample Output:

1 2 3 6 5 4

### Test Case 1

#### Input

10

6 7 8 4 3 2 1 5 9 10

#### Output

1 2 3 4 5 10 9 8 7 6

### C Program

```
#include<stdio.h>
void sortA(int m, int x[ ])
{
    int i, j, t;
    for(i = 1; i <= m-1; i++)
    {
        for(j = 1; j <= m-i; j++)
        {
            if(x[j-1] >= x[j])
            {
                t = x[j-1];
                x[j-1] = x[j];
                x[j] = t;
            }
        }
    }
}
void printorder(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n / 2; i++)
    {
        printf("%d ", arr[i]);
    }
    for (j = n - 1; j >= n / 2; j--)
    {
```

```

        printf("%d ", arr[j]);
    }
}
int main()
{
    int i, j, n, number[100];
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);
    sortA(n,number);
    printorder(number,n);
    return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

void sortA(int m, int x[ ])
{
    int i, j, t;
    for(i = 1; i <= m-1; i++)
    {
        for(j = 1; j <= m-i; j++)
        {
            if(x[j-1] >= x[j])
            {
                t = x[j-1];
                x[j-1] = x[j];
                x[j] = t;
            }
        }
    }
}

void printorder(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n / 2; i++)
    {
        printf("%d ", arr[i]);
    }
    for (j = n - 1; j >= n / 2; j--)
    {

```

```
        printf("%d ", arr[j]);
    }
}
int main()
{
    int i, j, n, number[100];
    cin>>n;
    for (i = 0; i < n; ++i)
        cin>>number[i];
    sortA(n,number);
    printorder(number,n);
    return 0;
}
```

## PROGRAM 47

The task is to find the minimum sum of Products of two arrays of the same size, given that k modifications are allowed on the first array. In each modification, one array element of the first array can either be increased or decreased by 2.

**Note:** the product sum is Summation ( $A[i]*B[i]$ ) for all i from 1 to n where n is the size of both arrays

**Input Format:** First line of the input contains n and k delimited by whitespace

Second line contains the Array A (modifiable array) with its values delimited by spaces

Third line contains the Array B (non-modifiable array) with its values delimited by spaces

**Output Format:** Output the minimum sum of products of the two arrays

### Constraints:

$$1 \leq N \leq 10^5$$

$$0 \leq |A[i]|, |B[i]| \leq 10^5$$

$$0 \leq K \leq 10^9$$

### Test Case 1

#### Input

5 3 2 3 4 5 4 3 4 2 3 2

#### Output

25

### Test Case 2

#### Input

3 5 1 2 -3 -2 3 -5

#### Output

-31

## C Program

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

```
#include <stdlib.h>
```

```
long int minproduct(long int a[], long int b[], long int n, long int k)
```

```
{
```

```
    long int diff = 0, res = 0;
```

```
    long int temp;
```

```
    for (long int i = 0; i < n; i++)
```

```
    {
```

```
        long int pro = a[i] * b[i];
```

```
        res = res + pro;
```

```
        if (pro < 0 && b[i] < 0) // b[i] is -ve
```

```
            temp = (a[i] + 2 * k) * b[i];
```

```

    else if (pro < 0 && a[i] < 0) // pro -ve and a[i] -ve
        temp = (a[i] - 2 * k) * b[i];

    else if (pro > 0 && a[i] < 0) // pro +ve and a[i] -ve
        temp = (a[i] + 2 * k) * b[i];

    else if (pro > 0 && a[i] > 0) // pro +ve and a[i] +ve
        temp = (a[i] - 2 * k) * b[i];

    long int d = abs(pro - temp);
    if (d > diff)
        diff = d;
}
return res - diff;
}
int main()
{
    long int n,mod;
    scanf("%ld %ld",&n,&mod);
    long int a[n], b[n];
    for (long int i = 0; i < n; i++)
    {
        scanf("%ld",&a[i]);
    }
    for (long int i = 0; i < n; i++)
    {
        scanf("%ld",&b[i]);
    }
    printf("%ld",minproduct(a, b, n, mod));
    return 0;
}

```

### C++ Program

```

#include <bits/stdc++.h>
using namespace std;
long int minproduct(long int a[], long int b[], long int n, long int k)
{
    long int diff = 0, res = 0;
    long int temp;
    for (long int i = 0; i < n; i++)
    {
        long int pro = a[i] * b[i];

```



```

    res = res + pro;
    if (pro < 0 && b[i] < 0) // b[i] is -ve
        temp = (a[i] + 2 * k) * b[i];

    else if (pro < 0 && a[i] < 0) // pro -ve and a[i] -ve
        temp = (a[i] - 2 * k) * b[i];

    else if (pro > 0 && a[i] < 0) // pro +ve and a[i] -ve
        temp = (a[i] + 2 * k) * b[i];

    else if (pro > 0 && a[i] > 0) // pro +ve and a[i] +ve
        temp = (a[i] - 2 * k) * b[i];

    long int d = abs(pro - temp);
    if (d > diff)
        diff = d;
}
return res - diff;
}
int main()
{
    long int n,mod;
    cin>>n>>mod;
    long int a[n], b[n];
    for (long int i = 0; i < n; i++)
    {
        cin>>a[i];
    }
    for (long int i = 0; i < n; i++)
    {
        cin>>b[i];
    }
    cout<<minproduct(a, b, n, mod);
    return 0;
}

```

## JAVA

```

import java.math.*;
import java.util.*;

```

```

class Main
{
    static int minproduct(int a[], int b[], int n, int k)

```

```

{
    int diff = 0, res = 0;
    int temp = 0;
    for (int i = 0; i < n; i++)
    {
        int pro = a[i] * b[i];
        res = res + pro;
        if (pro < 0 && b[i] < 0)
            temp = (a[i] + 2 * k) * b[i];

        else if (pro < 0 && a[i] < 0)
            temp = (a[i] - 2 * k) * b[i];

        else if (pro > 0 && a[i] < 0)
            temp = (a[i] + 2 * k) * b[i];

        else if (pro > 0 && a[i] > 0)
            temp = (a[i] - 2 * k) * b[i];

        int d = Math.abs(pro - temp);
        if (d > diff)
            diff = d;
    }
    return res - diff;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    int size, mod, count = 0;
    size = sc.nextInt();
    mod = sc.nextInt();
    int a[] = new int[size];
    int b[] = new int[size];
    for (int i = 0; i < size; i++)
    {
        a[i] = sc.nextInt();
    }
    for (int i = 0; i < size; i++)
    {
        b[i] = sc.nextInt();
    }
    System.out.println(minproduct(a, b, size, mod));
}
}

```

## Python

```
def minproduct(a,b,n,k):

    diff = 0
    res = 0
    for i in range(n):

        pro = a[i] * b[i]
        res = res + pro

        if (pro < 0 and b[i] < 0):
            temp = (a[i] + 2 * k) * b[i]

        elif (pro < 0 and a[i] < 0):
            temp = (a[i] - 2 * k) * b[i]

        elif (pro > 0 and a[i] < 0):
            temp = (a[i] + 2 * k) * b[i]

        elif (pro > 0 and a[i] > 0):
            temp = (a[i] - 2 * k) * b[i]

        d = abs(pro - temp)

        if (d > diff):
            diff = d
    return res - diff

n, k = [int(x) for x in input().split()]
a = [int(x) for x in input().split()]
b = [int(x) for x in input().split()]

print(minproduct(a, b, n, k))
```

## PROGRAM 48

Write a program to perform alternate sorting.

### Input:

Size of an array

List of characters

**Output:** array is alternately sorted

### Test Case 1

#### Input

5

5 1 4 7 9

#### Output

4 1 5 7 9

### C Program

```
#include<stdio.h>
#include<stdlib.h>
void fn_sorting(int *arr, int n)
{
    int temp, i, j;
    for(i=0; i<n;i=i+2)
    {
        for(j=i+2; j<n;j=j+2)
        {
            if(arr[i] > arr[j])
            {
                temp  = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    for(i=1; i<n;i=i+2)
    {
        for(j=i+2; j<n;j=j+2)
        {
            if(arr[i] > arr[j])
            {
                temp  = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```

```

    }
}
}
for(i=0; i<n; i++)
{
    printf("%d ", arr[i]);
}
}

int main()
{
    int num, i, j, *ptr;
    scanf("%d", &num);                // enter the number of elements
    ptr = (int*) malloc(num * sizeof(int));    //memory allocated using malloc
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    for(i = 0; i < num; i++)
    {
        scanf("%d", ptr + i);
    }
    fn_sorting(ptr, num);
    return 0;
}

```

## PROGRAM 49

Decode the logic and print the pattern that corresponds to the given input.

**Input Format:** Single Integer N.

**Output Format:** Output is the form of pattern.

**Constraints:**  $2 \leq N \leq 100$ ; Refer the sample output for formatting

### Test Case 1

#### Input

3

#### Output

10203010011012

\*\*4050809

\*\*\*\*607

### Test Case 2

#### Input

4

#### Output

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*10011

## C Program

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

```
int main() {
```

```
    int num, star=0;
```

```
    int space;
```

```
    scanf("%d",&num);
```

```
    int i, j, lterm, rterm;
```

```
    lterm = 1;
```

```
    rterm = num * num + 1;
```

```
    for (i = num; i > 0; i--)
```

```
    {
```

```
        for (space = 1; space <= star ; space++)
```

```
            printf( " ");
```

```
        star +=2;
```

```
        for (j = 1; j <= i; j++)
```

```
        {
```

```

        printf("%d0", lterm) ;
        lterm++;
    }
    for (j = 1; j <= i; j++)
    {
        printf("%d", rterm) ;
        if (j < i)
            printf( "0");
        rterm++;
    }
    rterm = rterm - (i - 1)*2 - 1;
    printf( "\n");
}
}

```

### C++ Program

```

#include <iostream>
using namespace std;

int main()
{

    int num, star=0;
    int space;
    cin>>num;
    int i, j, lterm, rterm;
    lterm = 1;
    rterm = num * num + 1;
    for (i = num; i > 0; i--)
    {
        for (space = 1; space <= star ; space++)
            cout << " ";
        star +=2;
        for (j = 1; j <= i; j++)
        {
            cout << lterm <<"0";
            lterm++;
        }
        for (j = 1; j <= i; j++)
        {
            cout << rterm;
            if (j < i)
                cout<<"0";

```

```

        rterm++;
    }
    rterm = rterm - (i - 1) * 2 - 1;
    cout << endl;
}
}

```

## JAVA program

```
import java.util.Scanner;
```

```
class Main
```

```

{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int num, star=0;
        int space;
        num = sc.nextInt();
        int i, j, lterm, rterm;
        lterm = 1;
        rterm = num * num + 1;
        for (i = num; i > 0; i--)
        {
            for (space = 1; space <= star ; space++)
                System.out.print("*");
            star +=2;
            for (j = 1; j <= i; j++)
            {
                System.out.print(lterm+"0");
                lterm++;
            }
            for (j = 1; j <= i; j++)
            {
                System.out.print(rterm);
                if (j < i)
                    System.out.print("0");
                rterm++;
            }
            rterm = rterm - (i - 1) * 2 - 1;
            System.out.println();
        }
    }
}

```



## Python

```
num = int(input())
star = 0
lterm = 1
rterm = num * num + 1
for i in range(num,0,-1):
    for space in range(1,star+1):
        print("*",sep="",end="")
    star = star + 2
    for j in range(1,i+1):
        print(lterm,"0",sep="",end="")
        lterm = lterm+1
    for j in range(1,i+1):
        print(rterm,sep="",end="")
    if (j < i) :
        print("0",end="")
    rterm=rterm+1
rterm = rterm - (i - 1) * 2 - 1
print()
```

## PROGRAM 50

Write a program to find out the prime factors of a given number.

**Input Format:** Input consists of integer

**Output Format:** Refer the sample output format

**Sample Input:**

10

**Sample Output:**

2

5

**Test Case 1**

**Input**

15

**Output**

3

5

## C Program

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

```

    }
}
return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;
int main()
{
    int i, j, num, isPrime;
    cin>>num;
    for(i=2; i<=num; i++)
    {
        if(num%i==0)
        {
            isPrime = 1;
            for(j=2; j<=i/2; j++)
            {
                if(i%j==0)
                {
                    isPrime = 0;
                    break;
                }
            }
            if(isPrime==1)
            {
                cout<< i<<endl;
            }
        }
    }
    return 0;
}

```

## PROGRAM 51

Our Friend Monk has finally found the Temple of Programming secrets. However, the door of the temple is firmly locked. Now, as per the rules of the temple, Monk needs to enter a Secret Password in a special language to unlock the door. This language, unlike English consists of K alphabets. The properties of this secret password are: It has a length of N characters. It is composed only of the K characters belonging to the Special language. Each character belonging to the special language has been used at max once in the secret code. Now, Monk has no idea about what the ideal password may be and needs you help. You need to help Monk find the total number of distinct candidate Strings for it Modulo  $10^9+7$ .

**Input Format:** The first line contains a single integer T denoting the number of test cases. Each of the next T lines contain two integers N and K denoting the length of the Secret Password and the number of characters of the Special language to be used respectively.

**Output Format:** For each test case, output the number of possible distinct secret passwords Modulo  $10^9+7$ .

### Constraints:

$$1 \leq T \leq 10$$

$$1 \leq N \leq K \leq 105$$

**Note:** You need to print the value of each element and not their weight.

### Sample Input:

1

3 3

### Output:

6

### Explanation:

Let's number the characters of the special language to be 1, 2 and 3 respectively. So, all possible candidate Strings are:

123

132

213

231

312

321

So, here we have 6 possible passwords. So, the answer =  $6\%(10^9+7)=6$ .

### Test Case 1

#### Input

2 4 21 30 300

#### Output

143640

87168287

## C++ Program

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
ll MOD = 1000000007;

int main()
{
    ll T,n,res,k,i;
    cin>>T;
    while(T--)
    {
        cin>>n>>k;
        res = 1;
        for(i=k-n+1;i<=k;i++)
            res = (res*i)%MOD;
        cout<<res<<"\n";
    }

    return 0;
}
```

## PROGRAM 52

Find the Ceiling value of a given number in the given Sorted Array. If ceiling value found, print the value else print value does not exist.

### Sample Input:

Size of array = 5

arr[] = {3, 6, 9, 12, 15}

ceiling value for : 4

### Sample Output:

6

### Test Case 1

#### Input

5

3

6

9

12

15

4

#### Output

6

### C Program

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

```
int ceilSearch(int arr[], int low, int high, int x)
```

```
{
    int i;
    if (x <= arr[low])
        return low;
    for (i = low; i < high; i++)
    {
        if (arr[i] == x)
            return i;

        if (arr[i] < x && arr[i + 1] >= x)
            return i + 1;
    }

    return -1;
}
```

```
int main()
```

```

{
    int arr[20],I;
    int n;
    float x;
    scanf("%d", &n);
    for(i = 0;i < n;i ++)
    {
        scanf("%d", &arr[i]);
    }
    scanf("%f", &x);
    int index = ceilSearch(arr, 0, n-1, x);
    if(index == -1){
        printf("Ceiling value doesn't exist");
    }
    else{
        printf("%d",arr[index]);
    }
    return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

```

```

int ceilSearch(int arr[], int low, int high, int x)
{
    int i;
    if (x <= arr[low])
        return low;
    for (i = low; i < high; i++)
    {
        if (arr[i] == x)
            return i;

        if (arr[i] < x && arr[i + 1] >= x)
            return i + 1;
    }

    return -1;
}

```

```

int main()
{

```

```

int arr[20],i;
int n;
float x;
cin>>n;
for(i = 0;i < n;i ++)
{
    cin>>arr[i];
}
cin>>x;
int index = ceilSearch(arr, 0, n-1, x);
if(index == -1){
    cout<<"Ceiling value doesn't exist";
}
else{
    cout<<arr[index];
}
return 0;
}

```



### PROGRAM 53

Raj and max wanted to test their students learning. Hence they come up with a problem. The problem consists of a string containing an integer. You should convert the string to its equivalent integer and then find the sum of prime number. There can be negative number also hence you need to multiply the negative value by 11. In case of invalid string print the output as 0.

You have been provided with a strict instruction that you should not use atoi() function in this problem. Usage of this function will refrain you from any upcoming scholarships.

**Input Format:** String contains the integer value.

**Output Format:** Should be an Integer.

#### Sample Input:

12

#### Sample output:

28

#### Explanation:

$2+3+5+7+11 = 28$

#### Test Case 1

##### Input

12

##### Output

28

#### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
int ATOI(string);
int primesum(int);
int main()
{
    string s;
    getline(cin,s);
    int val = ATOI(s);
    int result = primesum(val);
    cout<<result;
    return 0;
}
bool isNumericChar(char x)
{
    return (x >= '0' && x <= '9') ? true : false;
```

```

}
int ATOI(string str)
{
    if (str[0] == '\0')
        return 0;
    int res = 0;
    int sign = 1;
    int i = 0;
    if (str[0] == '-')
    {
        sign = -1;
        i++;
    }
    for (; str[i] != '\0'; ++i)
    {
        if (isNumericChar(str[i]) == false)
            return 0;
        res = res * 10 + str[i] - '0';
    }
    return sign * res;
}
int isprime(int n)
{
    int i, flag=1;
    for(i=2; i<=sqrt(n); i++)
    {
        if(n%i==0)
        {
            flag=0;
            break;
        }
    }
    return flag;
}
int primesum(int val)
{
    if(val<0)
    {
        return val*11;
    }
    int sum=0;
    for(int i=2; i<val; i++)
    {
        if(isprime(i))

```

```
{  
    sum+=i;  
}  
}  
return sum;  
}
```

## PROGRAM 54

Print the Words Ending with Letter S

This program takes a string as input and print the words ending with letter s.

### Input:

this was mine

### Output:

this

was

### Test Case 1

#### Input

kings are born at june

#### Output

kings

## C Program

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

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

```
char str[100];
```

```
void main()
```

```
{
```

```
    int i, t, j, len;
```

```
    scanf("%[^\n]s", str);
```

```
    len = strlen(str);
```

```
    str[len] = ' ';
```

```
    for (t = 0, i = 0; i < strlen(str); i++)
```

```
    {
```

```
        if ((str[i] == ' ') && (str[i - 1] == 's'))
```

```
        {
```

```
            for (j = t; j < i; j++)
```

```
            {
```

```
                printf("%c", str[j]);
```

```
            }
```

```
            t = i + 1;
```

```
            printf("\n");
```

```
        }
```

```

else
{
    if (str[i] == ' ')
    {
        t = i + 1;
    }
}
}
}

```

### C++ Program

```

#include <iostream>
#include <string.h>
using namespace std;

```

```

char str[100];

```

```

int main()

```

```

{

```

```

    int i, t, j, len;

```

```

    scanf("%[^\\n]s", str);

```

```

    len = strlen(str);

```

```

    str[len] = ' ';

```

```

    for (t = 0, i = 0; i < strlen(str); i++)

```

```

    {

```

```

        if ((str[i] == ' ') && (str[i - 1] == 's'))

```

```

        {

```

```

            for (j = t; j < i; j++)

```

```

            {

```

```

                printf("%c", str[j]);

```

```

            }

```

```

            t = i + 1;

```

```

            printf("\\n");

```

```

        }

```

```

    else

```

```

    {

```

```

        if (str[i] == ' ')

```

```

        {

```

```

            t = i + 1;

```

}

}

}

}

## PROGRAM 55

### Problem Description:

N number of people participated in a coding marathon where they were asked to solve some problems. Each problem carried 1 mark and at the end of the marathon, the total marks that each person achieved was calculated. As an organizer, you have the list of the total marks that each person achieved. You have to calculate the sum of the marks of top K scorers from the list.

### Input Specification:

input1: N, Total number of participants

input2: Top scorers

input3: An array of length N with the scores of all N participants

**Output Specification:** Return S, sum of the marks of top K scorers from the list.

### Example 1:

input1: 4

input2: 2

input3: {4, 1, 2, 5}

### Output:

9

### Explanation:

Top 2 scores are 5 and 4. Sum = 5+4=9.

### Example 2:

input1: 4

input2: 3

input3: {4, 3, 6, 1}

### Output:

13

### Explanation:

Top 3 scores are 6, 4 and 3. Sum = 6+4+3=13.

### Test Case 1

#### Input

4 2 4 1 2 5

#### Output

9

### Test Case 2

#### Input

4 3 4 3 6 1

#### Output

13

## C++ Program

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int N, K, sum = 0;
    cin >> N >> K;
    int arr[N];
    for(int i = 0; i < N; i++)
        cin >> arr[i];
    sort(arr, arr+N, greater<int>());

    for(int i = 0; i < K; i++)
        sum = sum + arr[i];

    cout << sum;
    return 0;
}
```



## PROGRAM 56

Sansa and Daenerys are close friends. They both are studying in the same school. Now they are on their summer vacation. As they are bored, they ask their parents to take them to an exhibition. There Sansa and Daenerys play a game. In this game, there are three boxes with some number written on their top. There is a treasure in one of the three boxes and the treasure is present in the box with the second largest number on its top. Also, to open the box, they need to decode the correct code of this box. The clue given to them to find the code is that it is the largest number which divides all the three given numbers. So, now help Sansa and Daenerys to decode the code.

### INPUT FORMAT:

Input consists of three integers. First input corresponds to the number of the first box. Second input corresponds to the number of the second box. Third input corresponds to the number of the third box.

#### Test Case 1

##### Input

2  
4  
6

##### Output

The treasure is in box which has number 4  
The code to open the box is 2

#### Test Case 2

##### Input

7  
5  
6

##### Output

The treasure is in box which has number 6  
The code to open the box is 1

#### Test Case 3

##### Input

3  
84  
15

##### Output

The treasure is in box which has number 15  
The code to open the box is 3

### C++ Program

```
#include<iostream>
using namespace std;
int main()
```

```

{
    int a,b,c,max=0,max1=0,val;
    cin>>a>>b>>c;
    if((a<b && a>c) || (a>b && a<c))
        cout<<"The treasure is in box which has number "<<a<<endl;
    else if((b>a && b<c) || (b>c && b<a))
        cout<<"The treasure is in box which has number "<<b<<endl;
    else if((c>a && c<b) || (c>b && c<a))
        cout<<"The treasure is in box which has number "<<c<<endl;
    if(a>b && a>c)
        max = a;
    else if(b>c)
        max = b;
    else
        max = c;
    for(int i=1;i<=max;i++)
    {
        if(a%i == 0 && b%i == 0 && c%i == 0)
            if(i>max1)
                max1 = i;
    }
    cout<<"The code to open the box is "<<max1;
}

```

## PROGRAM 57

Repeatedly find the sum of digits of a number until it becomes a single-digit number (say sod). Then compute the factorial of sod and print it.

**Input Format:** One line containing an integer "n" denoting number

**Output Format:** One number denoting the factorial of single digit

**Constraints:**  $1 \leq n \leq 10000$

### Sample Input

23

### Sample Output

120

### Test Case 1

#### Input

78

#### Output

720

### C Program

```
#include <stdio.h>
int sum_of_digits(int n);
int main()
{
    int N;
    scanf("%d", &N);
    int sod = N;
    while(sod >= 10)
    {
        sod = sum_of_digits(sod);
    }
    int fact = 1;
    for(int i = 1; i <= sod; i++)
    {
        fact = fact * i;
    }
    printf("%d", fact);
    return 0;
}
int sum_of_digits(int n)
{
```

```

int sum = 0;
while(n > 0)
{
    sum = sum + (n % 10);
    n = n / 10;
}
return sum;
}

```

### C++ Program

```

#include <iostream>
using namespace std;
int sum_of_digits(int n);
int main()
{
    int N;
    cin>>N;
    int sod = N;
    while(sod >= 10)
    {
        sod = sum_of_digits(sod);
    }
    int fact = 1;
    for(int i = 1; i <= sod; i++)
    {
        fact = fact * i;
    }
    cout<<fact;
    return 0;
}
int sum_of_digits(int n)
{
    int sum = 0;
    while(n > 0)
    {
        sum = sum + (n % 10);
        n = n / 10;
    }
    return sum;
}

```

## PROGRAM 58

It's your job to calculate the cost of replacing the damaged battle droids and to check whether it is within the budget limit of Rs.15000. The cost of the equipment and the parts are given below. Write a program to solve this problem.  
Blast Rifle Rs. 350.34 Visual Sensors Rs. 230.90 Auditory Sensors Rs. 190.55 Arms Rs. 125.30 Legs Rs. 180.90

**INPUT FORMAT:** Input consists of 5 integers.

The first input denotes the number of blast rifles to be replaced.

The second input denotes the number of visual sensors to be replaced.

The third input denotes the number of auditory sensors to be replaced.

The fourth input denotes the number of arms to be replaced.

The fifth input denotes the number of legs to be replaced.

**OUTPUT FORMAT:** If the total cost of replacing damaged battle droids is within the sanctioned budget of Rs. 15000, print "Yes". Otherwise, print "No". Refer the sample output for formatting.

### Test Case 1

#### Input

20

10

14

3

9

#### Output

Yes

### Test Case 2

#### Input

30

20

24

13

9

#### Output

No

### Test Case 3

#### Input

13

12

14

13

17

#### Output

Yes

## C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int blast,visual,audi,arms,legs;
    cin>>blast>>visual>>audi>>arms>>legs;
    int total =((blast*350.34) + (visual*230.90) + (audi*190.55) + (arms*125.30) + (legs*180.90));
    if(total <=15000)
    {
        cout<<"Yes";
    }
    else
    {
        cout<<"No";
    }
}
```

## PROGRAM 59

Write a program to return the nearest 10 multiple of any given number

### Sample Input:

18

### Sample Output:

20

### Test Case 1

#### Input

12

#### Output

10

### C Program

```
#include <stdio.h>
int main()
{
    int n,a;
    scanf("%d",&n);
    a=n%10;
    if(a<5)
        n=n-a;
    else
        n=n-a+10;
    printf("%d",n);
    return 0;
}
```

### C++ Program

```
#include <iostream>
using namespace std;

int main()
{
    int n,a;
    cin>>n;
    a=n%10;
    if(a<5)
        n=n-a;
    else
        n=n-a+10;
    cout<<n;
    return 0;
}
```

## PROGRAM 60

Write a program to check whether the number is Triangular or not. A number is termed as triangular number if we can represent it in the form of triangular grid of points such that the points form an equilateral triangle and each row contains as many points as the row number, i.e., the first row has one point, the second row has two points, the third row has three points and so on. The starting triangular numbers are 1, 3 (1+2), 6 (1+2+3), 10 (1+2+3+4).

**Input Format:** The first line of input contains an integer N denotes the number to be checked if it is triangular or not.

**Output Format:** If the number is Triangular then display 1 otherwise 0.

**Constraints:**  $1 \leq N \leq 10000000$

### Sample Input:

3

### Sample Output:

1

### Test Case 1

#### Input

10

#### Output

1

## C Program

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int sum = 0, is_found = 0;
    for(int i = 1; sum <= n; i++){
        sum = sum + i;
        if(sum == n){
            is_found = 1;
            break;
        }
    }
    if(is_found){
        printf("1\n");
    }
    else{
        printf("0\n");
    }
}
```



```
    return 0;
}
```

### **C++ Program**

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n;
    cin>>n;
    int sum = 0, is_found = 0;
    for(int i = 1; sum <= n; i++){
        sum = sum + i;
        if(sum == n){
            is_found = 1;
            break;
        }
    }
    if(is_found){
        cout<<"1\n";
    }
    else{
        cout<<"0\n";
    }
    return 0;
}
```

## PROGRAM 61

Write a program that calculates the day of the week for any particular date in the past or future.

**Input:** The first line of input contains a single integer T denoting the number of test cases. Then T test cases follow. Each test case consist of one line. The first line of each test case consists of an integer d,m and y separated by space. d is day, m is month and y is the year.

**Output:** Print day of given date.

### Constraints:

$1 \leq T \leq 100$

$1 \leq d \leq 31$

$1 \leq m \leq 12$

$1990 \leq Y \leq 2100$

### Test Case 1

#### Input

2 28 12 1995 30 8 2010

#### Output

Thursday

Monday

### Test Case 2

#### Input

1 24 07 2020

#### Output

Friday

## C Program

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

```
int main(){
```

```
    int date, month, year;
```

```
    scanf("%d%d%d", &date, &month, &year);
```

```
    int rem1 = date % 7;
```

```
    int rem2 = 0, rem3 = 0;
```

```
    if(year % 4 == 0){
```

```
        if(month == 1){
```

```
            rem2 = 5;
```

```
        }
```

```
        else if(month == 2){
```

```
            rem2 = 1;
```

```
        }
```

```

}
else{
    if(month == 1){
        rem2 = 6;
    }
    else if(month == 2){
        rem2 = 2;
    }
}
switch(month){
    case 10:
        rem2 = 6;
        break;
    case 3:
    case 11:
        rem2 = 2;
        break;
    case 4:
    case 7:
        rem2 = 5;
        break;
    case 6:
        rem2 = 3;
        break;
    case 8:
        rem2 = 1;
        break;
    case 9:
    case 12:
        rem2 = 4;
        break;
}
int r = year % 400;
if(r >= 100){
    int x = r % 100;
    int y = r / 100;
    int om = 5;
    while(y > 1){
        om = om + 5;
        y--;
    }
    om = om % 7;
    int m = om + x + (x/4);
    rem3 = m % 7;
}

```

```

}
else{
    int k = r + (r / 4);
    rem3 = k % 7;
}
int sum = rem1+rem2+rem3;
int z = sum % 7;
switch(z){
    case 0: printf("Sunday");
            break;
    case 1: printf("Monday");
            break;
    case 2: printf("Tuesday");
            break;
    case 3: printf("Wednesday");
            break;
    case 4: printf("Thursday");
            break;
    case 5: printf("Friday");
            break;
    case 6: printf("Saturday");
            break;
}
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main(){

    int date, month, year;
    cin>>date>>month>>year;
    int rem1 = date % 7;
    int rem2 = 0, rem3 = 0;
    if(year % 4 == 0){
        if(month == 1){
            rem2 = 5;
        }
        else if(month == 2){
            rem2 = 1;
        }
    }
    else{

```

```

    if(month == 1){
        rem2 = 6;
    }
    else if(month == 2){
        rem2 = 2;
    }
}
switch(month){
    case 10:
        rem2 = 6;
        break;
    case 3:
    case 11:
        rem2 = 2;
        break;
    case 4:
    case 7:
        rem2 = 5;
        break;
    case 6:
        rem2 = 3;
        break;
    case 8:
        rem2 = 1;
        break;
    case 9:
    case 12:
        rem2 = 4;
        break;
}
int r = year % 400;
if(r >= 100){
    int x = r % 100;
    int y = r / 100;
    int om = 5;
    while(y > 1){
        om = om + 5;
        y--;
    }
    om = om % 7;
    int m = om + x + (x/4);
    rem3 = m % 7;
}
else{

```

```

    int k = r + (r / 4);
    rem3 = k % 7;
}
int sum = rem1+rem2+rem3;
int z = sum % 7;
switch(z){
    case 0:
        cout<<"Sunday";
        break;
    case 1: cout<<"Monday";
        break;
    case 2: cout<<"Tuesday";
        break;
    case 3: cout<<"Wednesday";
        break;
    case 4: cout<<"Thursday";
        break;
    case 5: cout<<"Friday";
        break;
    case 6: cout<<"Saturday";
        break;
}
}

```

### **JAVA Program**

```

import java.util.Scanner;
class Main{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);

        int date = in.nextInt();
        int month = in.nextInt();
        int year = in.nextInt();
        int rem1 = date % 7;
        int rem2 = 0, rem3 = 0;
        if(year % 4 == 0){
            if(month == 1){
                rem2 = 5;
            }
            else if(month == 2){
                rem2 = 1;
            }
        }
    }
}

```

```

else{
    if(month == 1){
        rem2 = 6;
    }
    else if(month == 2){
        rem2 = 2;
    }
}
switch(month){
    case 10:
        rem2 = 6;
        break;
    case 3:
    case 11:
        rem2 = 2;
        break;
    case 4:
    case 7:
        rem2 = 5;
        break;
    case 6:
        rem2 = 3;
        break;
    case 8:
        rem2 = 1;
        break;
    case 9:
    case 12:
        rem2 = 4;
        break;
}
int r = year % 400;
if(r >= 100){
    int x = r % 100;
    int y = r / 100;
    int om = 5;
    while(y > 1){
        om = om + 5;
        y--;
    }
    om = om % 7;
    int m = om + x + (x/4);
    rem3 = m % 7;
}

```

```
else{
    int k = r + (r / 4);
    rem3 = k % 7;
}
int sum = rem1+rem2+rem3;
int z = sum % 7;
switch(z){
    case 0: System.out.println("Sunday");
        break;
    case 1: System.out.println("Monday");
        break;
    case 2: System.out.println("Tuesday");
        break;
    case 3: System.out.println("Wednesday");
        break;
    case 4: System.out.println("Thursday");
        break;
    case 5: System.out.println("Friday");
        break;
    case 6: System.out.println("Saturday");
        break;
}
}
}
```



## PROGRAM 62

Dinesh is fond of video games. Due to the pandemic, he designs a video game called the Corona world. In this game, the player enters the game with a certain energy. The player should defeat all the corona infected zombies to reach the next level. When time increases the zombies will increase double the previous minute. Anyhow the player can manage to fight against all the zombies. In this case, definitely the player can not achieve the promotion. Hence the player gets a superpower to destroy all the zombies in the current level when the current game time is a palindrome. Anyhow the player can manage only if he knows the time taken to get the superpower. Help the player by providing the minimum minutes needed to get the superpower by which he can destroy all the zombies. You will be provided with the starting time of the game.

**Input Format:** First-line contains a string representing the starting time.

**Output:** A string representing the minimum minutes needed to get the superpower.

**Constraints:** Input time will be in 24 hours format

**Sample Input:**

05:39

**Sample Output:**

11

**Explanation:**

It takes 11 minutes for minute value to become 50, 05:50 is a palindromic time.

### Test Case 1

**Input**

13:31

**Output**

0

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
int get_palindrome_time(string str)
{
    int hh, mm;
    hh = (str[0] - 48) * 10 + (str[1] - 48);
    mm = (str[3] - 48) * 10 + (str[4] - 48);
    int requiredTime = 0;
    while (hh % 10 != mm / 10 || hh / 10 != mm % 10)
    {
        ++mm;
        if (mm == 60)
```

```

    {
        mm = 0;
        ++hh;
    }
    if (hh == 24)
        hh = 0;
    ++requiredTime;
}
return requiredTime;
}
int main()
{
    string str;
    getline(cin,str);
    cout << get_palindrome_time(str) << endl;
}

```

### PROGRAM 63

Sherlock Being tired with the usual coding rounds started growing his interest in reverse coding when he won the event in his college symposium. He wondered if his friend has the brain to quickly identify the pattern and verify if his inputs are correct or not. From the example portion given below, where you will be given a number(n) and its output, Using this find the pattern.

**Example to identify the pattern:**

Input	Output
10	55
20	210
5	15
0	0
1	1
2	3

Your task is that from the pattern you identified above, You have to tell if for the given n whether the given m is the correct answer or not.

**Input Format:** The first line consists of n and m.

**Output Format:** For each n and m output 1 if m is the corresponding output for the value of n and otherwise 0.

**Constraints:**

$0 \leq n \leq 1000$

$0 \leq m \leq 10^6$

**Sample Input - 1:**

6 21

**Sample Output - 1:**

1

**Sample Input - 2:**

4 11

**Sample Output - 2:**

0

**Test Case 1**

**Input**

383 73536

**Output**

1

### **C Program**

```
#include <stdio.h>

int main() {
    int n, p;
    scanf("%d %d", &n, &p);
    int sum = 0;
    for(int num = 1; num <= n; num++){
        sum = sum + num;
    }
    if(p == sum){
        printf("1");
    }
    else{
        printf("0");
    }
    return 0;
}
```

### **C++ Program**

```
#include <iostream>
using namespace std;

int main() {
    int n, p;
    scanf("%d %d", &n, &p);
    int sum = 0;
    for(int num = 1; num <= n; num++){
        sum = sum + num;
    }
    if(p == sum){
        printf("1");
    }
    else{
        printf("0");
    }
    return 0;
}
```

## PROGRAM 64

Everyone has some habits to collect one thing or the other. Harshit also has the craze to collect pens but in 3 piles. In the first pile, he collected A pens and in the second pile, he collected B pens but in the third and the last pile, he thinks of something different. He decided to collect only the minimum number of pens in the third pile so that the sum of pens in the three piles will give him a prime number.

Note: There should be at least one pen in the third pile.

### Input Format:

First input contains an integer A denotes the number of pens in pile A

Second input contains an integer B denotes the number of pens in pile B

**Output Format:** Print the minimum number of pens that should be there in the third pile so that sum of all three piles produces a prime number.

### Constraints:

$1 \leq A \leq 1000$

$1 \leq B \leq 1000$

### Sample Input:

1 3

### Sample Output

1

### Explanation:

Harshit put one pen in first pile and 3 pens in second pile which give 4 as a sum. So if he adds one pen in the third pile, the sum will yield a prime number ie.5

Therefore the answer is 1.

### Test Case 1

#### Input

4 3

#### Output

4

### C Program

```
#include<stdio.h>
#define MAX 1000
int is_prime(int n);
int main(){
    int A, B;
    scanf("%d%d", &A, &B);
    int sum = A + B;
    int i;
```

```

for(i = 1; i<=MAX; i++){
    sum = sum + 1;
    if(is_prime(sum)){
        break;
    }
}
printf("%d", i);
}
int is_prime(int n){
    int is_prime = 1;
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            is_prime = 0;
            break;
        }
    }
    return is_prime;
}

```

### C++ Program

```

#include <iostream>
using namespace std;
#define MAX 1000
int is_prime(int n);
int main(){
    int A, B;
    cin>>A>>B;
    int sum = A + B;
    int i;
    for(i = 1; i<=MAX; i++){
        sum = sum + 1;
        if(is_prime(sum)){
            break;
        }
    }
    cout<<i;
    return 0;
}
int is_prime(int n){
    int is_prime = 1;
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            is_prime = 0;

```

```
        break;
    }
}
return is_prime;
}
```

## PROGRAM 65

Write a program to sort the elements in an odd position in descending order and elements in even position in ascending order

### Test Case 1

#### Input

7  
13  
2  
4  
15  
12  
10  
5

#### Output

13 2 12 10 5 15 4

### C Program

```
#include<stdio.h>
int main(){
    int n, arr[100];
    scanf("%d", &n);
    for(int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }
    for(int i = 0; i < n; i=i+2){
        if(i % 2 == 0){
            for(int j = i; j < n; j = j + 2){
                if(arr[i] < arr[j]){
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
    for(int i = 1; i < n; i = i + 2){
        for(int j = i; j < n; j = j + 2){
            if(arr[i] > arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
```



```

    }
}
for(int i = 0; i < n; i++){
    printf("%d ", arr[i]);
}
}

```

## C++ Program

```

#include <iostream>
using namespace std;

```

```

int main(){
    int n, arr[100];
    cin>>n;
    for(int i = 0; i < n; i++){
        cin>>arr[i];
    }
    for(int i = 0; i < n; i=i+2){
        if(i % 2 == 0){
            for(int j = i; j < n; j = j + 2){
                if(arr[i] < arr[j]){
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
    for(int i = 1; i < n; i = i + 2){
        for(int j = i; j < n; j = j + 2){
            if(arr[i] > arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    for(int i = 0; i < n; i++){
        cout<<arr[i]<<" ";
    }
}

```

## PROGRAM 66

Given an array of positive integers. Your task is to find the leaders in the array.

Note: An element of array is leader if it is greater than or equal to all the elements to its right side. Also, the rightmost element is always a leader.

**Input:** The first line of 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 size of array. The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.

**Output:** Print all the leaders.

### Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 107$

$0 \leq A_i \leq 107$

### Input:

3  
6  
16 17 4 3 5 2

5  
1 2 3 4 0

5  
7 4 5 7 3

### Output:

17 5 2  
4 0  
7 7 3

### Explanation:

Testcase 3: All elements on the right of 7 (at index 0) are smaller than or equal to 7. Also, all the elements of right side of 7 (at index 3) are smaller than 7. And, the last element 3 is itself a leader since no elements are on its right.

### Test Case 1

#### Input

1 15 12 45 85 96 96 85 324 422 10 14 721 741 852 65 453

#### Output

852 453

### C Program

```
#include<stdio.h>
int main()
{
```

```

int t;
scanf("%d",&t);
while(t--)
{
    int i,n,len;
    scanf("%d",&n);
    int a[n],leader[n],count=0;
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    leader[count++]=a[n-1];
    for(i=n-2;i>=0;i--)
    {
        if(a[i]>=leader[count-1])
            leader[count++]=a[i];
    }
    for(i=count-1;i>=0;i--){
        printf("%d ",leader[i]);
    }
    printf("\n");
}
}

```

### C++ Program

```

#include<iostream>
#include<vector>
#define ll long long
using namespace std;
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int i,n,len;
        cin>>n;
        int a[n];
        for(i=0;i<n;i++)
            cin>>a[i];
        vector<int>st;
        st.push_back(a[n-1]);
        for(i=n-2;i>=0;i--){
            if(a[i]>=st.back())
                st.push_back(a[i]);
        }
    }
}

```

```
    }  
    len=st.size();  
    for(i=len-1;i>=0;i--){  
        cout<<st[i]<<" ";  
    }  
    cout<<endl;  
}  
return 0;  
}
```

## PROGRAM 67

A man his driving car from home to the office with X petrol. There is an N number of petrol bunks in the city with only a few capacities and each petrol is located in different places. For one km one litre will consume. So he fills up petrol in his petrol tank in each petrol bunks. Output the remaining petrol if he has or tells him that he cannot travel if he is out of petrol. Write a program for the above-mentioned scenario.

### Input Format:

The first input contains an integer which denotes the petrol in car

The second input contains an integer 'n' which denotes the number of petrol bunks

Get the 'n' number of petrol bunks separated by spaces

Get the 'n' number of distance for each petrol bunk

Get the 'n' number of petrol capacities for each petrol bunk

**Output Format:** Print the remaining petrol left in the car

### Sample Input:

Petrol in car: 2 Liters

Number of petrol bunks: 3

Petrol bunks: A B C

Distance from each petrol bunk: 1 5 3

Capacities of each petrol bunk: 6 4 2

### Sample Output:

5

### Explanation:

Remaining petrol in car is 5 liters

### Test Case 1

#### Input

2

3

a b c

1 5 3

6 4 2

#### Output

5

### Test Case 2

#### Input

2

3

A B C

1 5 3

2 3 1

## Output

No Petrol

## C Program

```
#include<stdio.h>
int main(){
    int petrol_capacity, n;
    scanf("%d", &petrol_capacity);
    scanf("%d\n", &n);
    char petrol_bunk[n];
    for(int i = 0; i < n; i++){
        scanf("%c\n",&petrol_bunk[i]);
    }
    int distance[n], capacities[n];
    for(int i = 0; i < n; i++){
        scanf("%d", &distance[i]);
    }
    for(int i = 0; i < n; i++){
        scanf("%d", &capacities[i]);
    }
    for(int i = 0; i < n; i++){
        if(petrol_capacity > 0){
            petrol_capacity = (petrol_capacity - distance[i]) + capacities[i];
        }
        else{
            break;
        }
    }
    if(petrol_capacity > 0){
        printf("%d", petrol_capacity);
    }
    else {
        printf("No Petrol");
    }
}
```

## C++ Program

```
include <iostream>
using namespace std;

int main(){
    int petrol_capacity, n;
```

```

cin>>petrol_capacity;
cin>>n;
char petrol_bunk[n];
for(int i = 0; i < n; i++){
    cin>>petrol_bunk[i];
}
int distance[n], capacities[n];
for(int i = 0; i < n; i++){
    cin>>distance[i];
}
for(int i = 0; i < n; i++){
    cin>>capacities[i];
}
for(int i = 0; i < n; i++){
    if(petrol_capacity > 0){
        petrol_capacity = (petrol_capacity - distance[i]) + capacities[i];
    }
    else{
        break;
    }
}
if(petrol_capacity > 0){
    cout<<petrol_capacity;
}
else {
    cout<<"No Petrol";
}
}

```

## PROGRAM 68

Given two positive numbers x and y, check if y is a power of x or not.

### Input Format:

The first input contains an integer which denotes the value of x

The second input contains an integer which denotes the value of y

**Output Format:** Print 1 if y is a power of x, else print 0.

### Constraints:

$1 \leq x \leq 103$

$1 \leq y \leq 108$

### Sample Input:

2

8

### Sample Output:

1

### Test Case 1

#### Input

1 3

#### Output

0

## C Program

```
#include<stdio.h>
#define MAX 1000
int main(){
    int b, p;
    scanf("%d%d", &b, &p);
    int is_found = 0;
    int n = 1;
    for(int i = 1; i <= MAX; i++)
    {
        n = b * n;
        if(n == p || p == 1){
            is_found = 1;
            break;
        }
    }
    if(is_found == 1){
        printf("%d", is_found);
    }
```



```

    }
    else{
        printf("%d", is_found);
    }
}

```

### C++ Program

```

#include <iostream>
using namespace std;

#define MAX 1000
int main(){
    int b, p;
    cin>>b>>p;
    int is_found = 0;
    int n = 1;
    for(int i = 1; i <= MAX; i++)
    {
        n = b * n;
        if(n == p || p == 1){
            is_found = 1;
            break;
        }
    }
    if(is_found == 1){
        cout<<is_found;
    }
    else{
        cout<< is_found;
    }
}

```

## PROGRAM 69

Given a string S consisting only '0's and '1's, print the last index of the '1' present in it.

**Input:** First line of the input contains the number of test cases T, T lines follow each containing a stream of characters.

**Output:** Corresponding to every test case, output the last index of 1. If 1 is not present, print "-1" (without quotes).

### Constraints:

$1 \leq T \leq 110$

$1 \leq |S| \leq 10^6$

### Input:

2

00001

0

### Output:

4

-1

### Explanation:

Testcase 1: Last index of 1 in given string is 4.

Testcase 2: Since, 1 is not present, so output is -1.

### Test Case 1

#### Input

2 00000 11111

#### Output

-1

4

### C++ Program

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int t;
```

```
    cin >> t;
```

```
    for(int q=0;q<t;q++)
```

```
    {
```

```
        int ind=-1;
```

```
        string s;
```

```
cin >> s;  
for(int i=0;i<s.length();i++)  
{  
    if(s.at(i)=='1')  
        ind=i;  
}  
cout << ind << endl;  
}  
return 0;  
}
```

## PROGRAM 70

Write a program to calculate the sum of primes present in the digits of the given number.

**Input Format:** The input contains an integer N

**Output Format:** Print sum of primes in the digits of the given number

**Sample Input:**

333

**Sample Output:**

9

**Test Case 1**

**Input**

345

**Output**

8

## C Program

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    int sum = 0;
    while(n > 0){
        int r = n % 10;
        if(is_prime_number(r)){
            sum = sum + r;
        }
        n = n / 10;
    }
    printf("%d", sum);
}
int is_prime_number(int n){
    int is_prime = 1;
    if(n == 0 || n == 1){
        return 0;
    }
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            is_prime = 0;
            break;
        }
    }
```

```

    }
    return is_prime;
}

```

### C++ Program

```

#include <iostream>
using namespace std;
int is_prime_number(int);
int main(){
    int n;
    cin>>n;
    int sum = 0;
    while(n > 0){
        int r = n % 10;
        if(is_prime_number(r)){
            sum = sum + r;
        }
        n = n / 10;
    }
    cout<< sum;
}
int is_prime_number(int n){
    int is_prime = 1;
    if(n == 0 || n == 1){
        return 0;
    }
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            is_prime = 0;
            break;
        }
    }
    return is_prime;
}

```

### JAVA Program

```

import java.util.Scanner;
class Main{
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int sum = 0;
    }
}

```

```

while(n > 0){
    int r = n % 10;
    if(is_prime_number(r)){
        sum = sum + r;
    }
    n = n / 10;
}
System.out.println(sum);
}
public static boolean is_prime_number(int n){
    boolean is_prime = true;
    if(n == 0 || n == 1){
        return false;
    }
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            is_prime = false;
            break;
        }
    }
    return is_prime;
}
}

```

## PROGRAM 71

Given a binary array A[] of size N. The task is to arrange the array in increasing order.

**Note:** The binary array contains only 0 and 1.

**Input:** The first line of input contains an integer T, denoting the test cases. Every test case contains two lines, the first line is N(size of the array) and the second line is space-separated elements of the array.

**Output:** Space-separated elements of sorted arrays. There should be a new line between the output of every test case.

**Your Task:** Complete the function SortBinaryArray() which takes given array as input and returns the sorted array. Expected Time Complexity: O(N). Expected Auxiliary Space: O(1). Challenge: Try doing it in one pass.

### Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 10^6$

$0 \leq A[i] \leq 1$

### Test Case 1

#### Input

2 5 1 0 1 1 0 10 10 1 1 1 1 1 0 0 0

#### Output

0 0 1 1 1

0 0 0 0 1 1 1 1 1 1

### Test Case 2

#### Input

2 10 1 0 0 0 1 0 1 1 0 1 7 1 1 1 1 0 0 0

#### Output

0 0 0 0 0 1 1 1 1 1

0 0 0 1 1 1 1

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
vector<int> SortBinaryArray(vector<int> binArray);
vector<int> SortBinaryArray(vector<int> binArray)
{
    int z=0,o=0,i;
    for(i=0;i<binArray.size();i++){
        if(binArray[i]==0)z++;
        if(binArray[i]==1)o++;
    }
}
```

```

for(i=0;i<z;i++){
    binArray[i]=0;
}
for(i=z;i<binArray.size();i++){
    binArray[i]=1;
}
return binArray;
}

int main() {
    int t;
    cin>>t;

    while(t-->0)
    {
        int n;
        cin>>n;
        vector<int> binArray(n);

        for(int i = 0; i < n; i++)
            cin>>binArray[i];

        vector<int> result = SortBinaryArray(binArray);
        for(int i=0; i<n; i++)
            cout<<result[i]<<" ";
        cout<<endl;
    }
    return 0;
}

```



## PROGRAM 72

Given a set of numbers like {10, 36, 54,89,12}, find the sum of weights based on the following conditions: 1. Weight is 5 if a number is a perfect square 2. Weight is 4 if a number is multiple of 4 and divisible by 6 3. Weight is 3 if a number is even number

Sort the numbers and print it as follows

<10,its\_weight> <36,its\_weight> <89,its\_weight>

**Note:** Display the numbers based on increasing order.

### Test Case 1

#### Input

5

10 36 54 89 12

#### Output

<10,3> <12,7> <36,12> <54,3> <89,0>

### C Program

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

    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            if(arr[i] > arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    int weight[n];
    for(int i = 0; i < n; i++){
        weight[i] = 0;
        int s = (int)(sqrt(arr[i]));
        if(s * s == arr[i]){
            weight[i] = weight[i] + 5;
        }
        if(arr[i] % 4 == 0 && arr[i] % 6 == 0){
```

```

        weight[i] = weight[i] + 4;
    }
    if(arr[i] % 2 == 0){
        weight[i] = weight[i] + 3;
    }
}
for(int i = 0; i < n; i++){
    printf("<%d,%d> ", arr[i], weight[i]);
}
}

```

## C++ Program

```

#include <iostream>
using namespace std;

#include<math.h>
int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i = 0; i < n; i++){
        cin>>arr[i];
    }

    for(int i = 0; i < n; i++){
        for(int j = i + 1; j <n; j++){
            if(arr[i] > arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    int weight[n];
    for(int i = 0; i < n; i++){
        weight[i] = 0;
        int s = (int)(sqrt(arr[i]));
        if(s * s == arr[i]){
            weight[i] = weight[i] + 5;
        }
        if(arr[i] % 4 == 0 && arr[i] % 6 == 0){
            weight[i] = weight[i] + 4;
        }
    }
}

```

```

        if(arr[i] % 2 == 0){
            weight[i] = weight[i] + 3;
        }
    }
    for(int i = 0; i < n; i++){
        printf("<%d,%d> ", arr[i], weight[i]);
    }
}

```

## JAVA Program

```

import java.util.Scanner;
class Main{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int arr[] = new int[n];
        for(int i = 0; i < n; i++){
            arr[i] = in.nextInt();
        }
        for(int i = 0; i < n; i++){
            for(int j = i + 1; j < n; j++){
                if(arr[i] > arr[j]){
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        int weight[] = new int[n];
        for(int i = 0; i < n; i++){
            weight[i] = 0;
            int s = (int)(Math.sqrt(arr[i]));
            if(s * s == arr[i]){
                weight[i] = weight[i] + 5;
            }
            if(arr[i] % 4 == 0 && arr[i] % 6 == 0){
                weight[i] = weight[i] + 4;
            }
            if(arr[i] % 2 == 0){
                weight[i] = weight[i] + 3;
            }
        }
        for(int i = 0; i < n; i++){

```

```
        System.out.print("<" + arr[i] + "," + weight[i] + "> ");  
    }  
}  
}
```

## PROGRAM 73

Palindromes are strings that read the same backwards as forwards. Eg: aca, racecar etc You are given a string N, which is a number. You are allowed to make up to X number of alterations to the string. Your mission, should you choose to accept it, is to modify N to create the largest palindromic number possible without altering the length of the string N. Note: 1. The 0's at the left of the string N should also be considered in the length of N and in the palindrome. Eg: 011 cannot be considered as 11 and is not a palindrome, but 0110 is a palindrome. 2. An alteration is modifying a digit in the string N. Eg: 911 is obtained by making 1 alteration to 011.

### Input Format

The first line contains two space separated integers L and X, denoting the length of the string N and the maximum number of alterations allowed.

The second line contains an L-digit string of numbers that form N.

**Output Format:** Print a single line with the largest palindromic number formed by making up to X alterations to N. If a palindrome cannot be formed by making X alterations to N, print -1. Refer the sample output for formatting

### Test Case 1

#### Input

4 1  
3943

#### Output

3993

### Test Case 2

#### Input

5 1  
12345

#### Output

-1

### C++ Program

```
#include <bits/stdc++.h>
using namespace std;
```

```
string maximumPalinUsingKChanges(string str, int k)
{
    string palin = str;

    int l = 0;
    int r = str.length() - 1;

    // first try to make string palindrome
    while (l < r)
```

```

{
    // Replace left and right character by
    // maximum of both
    if (str[l] != str[r])
    {
        palin[l] = palin[r] = max(str[l], str[r]);
        k--;
    }
    l++;
    r--;
}

// If k is negative then we can't make
// string palindrome
if (k < 0)
    return "-1";
l = 0;
r = str.length() - 1;

while (l <= r)
{
    // At mid character, if K>0 then change
    // it to 9
    if (l == r)
    {
        if (k > 0)
            palin[l] = '9';
    }

    // If character at lth (same as rth) is
    // less than 9
    if (palin[l] < '9')
    {
        /* If none of them is changed in the
        previous loop then subtract 2 from K
        and convert both to 9 */
        if (k >= 2 && palin[l] == str[l] &&
            palin[r] == str[r])
        {
            k -= 2;
            palin[l] = palin[r] = '9';
        }

        /* If one of them is changed in the previous

```

```

        loop then subtract 1 from K (1 more is
        subtracted already) and make them 9 */
    else if (k >= 1 && (palin[l] != str[l] ||
        palin[r] != str[r]))
    {
        k--;
        palin[l] = palin[r] = '9';
    }
}
l++;
r--;
}
return palin;
}

```

```

string convertToString(char* a, int size)
{
    int i;
    string s = "";
    for (i = 0; i < size; i++) {
        s = s + a[i];
    }
    return s;
}

```

```

int main()
{
    int l,k;
    cin>>l>>k;
    char str1[l];
    cin>>str1;
    string str = convertToString(str1,l);
    cout << maximumPalinUsingKChanges(str, k);
    return 0;
}

```

## PROGRAM 74

Write a program to capitalize the first character of all the words in the given string

**Input Format:** One line containing the string

**Output Format:** String denoting capitalize the first character

### Sample Input

hello programmer

### Sample Output

Hello Programmer

### Test Case 1

#### Input

welcome to india

#### Output

Welcome To India

### C Program

```
#include<stdio.h>
int main()
{
    char str[26];
    scanf("%[^\\n]s", str);
    int i;
    for(i = 0; str[i]!='\\0'; i++)
    {
        if(i == 0)
        {
            if(str[i] >= 'a' && str[i] <= 'z')
            {
                str[i] = str[i] - 32;
            }
        }
        else if(str[i] == ' ')
        {
            if(str[i+1] >= 'a' && str[i+1] <= 'z')
            {
                str[i+1] = str[i+1] - 32;
            }
        }
    }
    printf("%s",str);
}
```



```
    return 0;
}
```

### **C++ Program**

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char str[26];
```

```
    scanf("%[^\\n]s", str);
```

```
    int i;
```

```
    for(i = 0; str[i]!='\\0'; i++)
```

```
    {
```

```
        if(i == 0)
```

```
        {
```

```
            if(str[i] >= 'a' && str[i] <= 'z')
```

```
            {
```

```
                str[i] = str[i] - 32;
```

```
            }
```

```
        }
```

```
        else if(str[i] == ' ')
```

```
        {
```

```
            if(str[i+1] >= 'a' && str[i+1] <= 'z')
```

```
            {
```

```
                str[i+1] = str[i+1] - 32;
```

```
            }
```

```
        }
```

```
    }
```

```
    cout<<str;
```

```
    return 0;
```

```
}
```

## PROGRAM 75

Given an integer n, write a program that returns "yes" if binary representation of n is palindrome else return "no".

### Sample Input:

9

### Sample Output:

yes

### Explanation:

The binary representation of 9 is 1001 and the binary representation is a palindrome so the output is yes.

### Test Case 1

#### Input

98

#### Output

no

### Test Case 2

#### Input

33

#### Output

yes

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int n1=0;
    int temp=n;
    while(n>0)
    {
        n1<<=1;
        n1|=(n&1);
        n>>=1;
    }
    if(temp == n1)
        cout<<"yes";
    else
        cout<<"no";
}
```

## PROGRAM 76

Write a program to print all the combinations of the given word with or without meaning (when unique characters are given).

### Test Case 1

#### Input

abc

#### Output

abc

acb

bac

bca

cba

cab

### C++ Program

```
#include<bits/stdc++.h>
using namespace std;
void permute(string a, int l, int r)
{
    if (l == r)
        cout<<a<<endl;
    else
    {
        for (int i = l; i <= r; i++)
        {
            swap(a[l], a[i]);
            permute(a, l+1, r);
            swap(a[l], a[i]);
        }
    }
}
int main()
{
    string str;
    cin>>str;
    int n = str.size();
    permute(str, 0, n-1);
    return 0;
}
```

## PROGRAM 77

Write a program to generate the following series 9, 11, 20, 31,..., 82

**INPUT & OUTPUT FORMAT:** The input is an integer which denotes 'n'. Print the series and refer the sample output for formatting.

### SAMPLE INPUT:

6

### SAMPLE OUTPUT:

9 11 20 31 51 82

### Test Case 1

#### Input

1

#### Output

9

### Test Case 2

#### Input

10

#### Output

9 11 20 31 51 82 133 215 348 563

### Test Case 3

#### Input

5

#### Output

9 11 20 31 51

## C++ Program

```
#include <iostream>
using namespace std;
#include<math.h>
int main()
{
    int n;
    cin>>n;
    int j=9,k=11,l=0;
    for(int i=1;i<=n;i++)
    {
        if(i==1)
        {
            cout<<j<<" ";
```

```
    }  
    else if(i==2)  
    {  
        cout<<k<<" ";  
    }  
    else  
    {  
        l=j+k;  
        cout<<l<<" ";  
        j=k;  
        k=l;  
    }  
}  
return 0;  
}
```

## PROGRAM 78

Write a program to merge given two sorted arrays such that the elements are not repeated

### Sample Input:

Size: 8

Array 1: 2, 4, 5, 6, 7, 9, 10, 13

Size: 10

Array 2: 2, 3, 4, 5, 6, 7, 8, 9, 11, 15

### Sample Output:

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15

### Test Case 1

#### Input

8

2 4 5 6 7 9 10 13

10

2 3 4 5 6 7 8 9 11 15

#### Output

2 3 4 5 6 7 8 9 10 11 13 15

### C Program

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

```
int main() {  
    int n1;  
    scanf("%d", &n1);  
    int arr1[n1];  
    for(int i = 0; i < n1; i++){  
        scanf("%d", &arr1[i]);  
    }  
    int n2;  
    scanf("%d", &n2);  
    int arr2[n2];  
    for(int i = 0; i < n2; i++){  
        scanf("%d", &arr2[i]);  
    }  
    int temp[n1 + n2];  
    int i = 0, j = 0, k = 0;  
    while(i < n1 && j < n2){  
        if(arr1[i] == arr2[j]){  
            temp[k++] = arr1[i];  
            i++;  
            j++;  
        }  
    }
```

```

        else if(arr1[i] < arr2[j]){
            temp[k++] = arr1[i];
            i++;
        }
        else if(arr2[j] < arr1[i]){
            temp[k++] = arr2[j];
            j++;
        }
    }
    while(i < n1){
        temp[k++] = arr1[i];
        i++;
    }
    while(j < n2){
        temp[k++] = arr2[j];
        j++;
    }
    for(int x = 0; x < k; x++){
        printf("%d ", temp[x]);
    }
    return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

int main() {
    int n1;
    cin>>n1;
    int arr1[n1];
    for(int i = 0; i < n1; i++){
        cin>>arr1[i];
    }
    int n2;
    cin>>n2;
    int arr2[n2];
    for(int i = 0; i < n2; i++){
        cin>>arr2[i];
    }
    int temp[n1 + n2];
    int i = 0, j = 0, k = 0;
    while(i < n1 && j < n2){

```

```

    if(arr1[i] == arr2[j]){
        temp[k++] = arr1[i];
        i++;
        j++;
    }
    else if(arr1[i] < arr2[j]){
        temp[k++] = arr1[i];
        i++;
    }
    else if(arr2[j] < arr1[i]){
        temp[k++] = arr2[j];
        j++;
    }
}
while(i < n1){
    temp[k++] = arr1[i];
    i++;
}
while(j < n2){
    temp[k++] = arr2[j];
    j++;
}
for(int x = 0; x < k; x++){
    cout<<temp[x]<<" ";
}
return 0;
}

```

## JAVA Program

```

import java.util.Scanner;
class Main{
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);
        int n1 = in.nextInt();
        int arr1[] = new int[n1];
        for(int i = 0; i < n1; i++){
            arr1[i] = in.nextInt();
        }
        int n2 = in.nextInt();
        int arr2[] = new int[n2];
        for(int i = 0; i < n2; i++){
            arr2[i] = in.nextInt();
        }
    }
}

```



```

int temp[] = new int[n1 + n2];
int i = 0, j = 0, k = 0;
while(i < n1 && j < n2){
    if(arr1[i] == arr2[j]){
        temp[k++] = arr1[i];
        i++;
        j++;
    }
    else if(arr1[i] < arr2[j]){
        temp[k++] = arr1[i];
        i++;
    }
    else if(arr2[j] < arr1[i]){
        temp[k++] = arr2[j];
        j++;
    }
}
while(i < n1){
    temp[k++] = arr1[i];
    i++;
}
while(j < n2){
    temp[k++] = arr2[j];
    j++;
}
for(int x = 0; x < k; x++){
    System.out.print(temp[x] + " ");
}
}

```

## PROGRAM 79

Given an array of integers where every element appears even number of times except one element which appears odd number of times, write a program to find that odd occurring element in  $O(\log n)$  time. The equal elements must appear in pairs in the array but there cannot be more than two consecutive occurrences of an element.

### For example:

3

2 3 2

It doesn't have equal elements appear in pairs

7

1 1 2 2 2 3 3

It contains three consecutive instances of an element.

### Test Case 1

#### Input

5

2 2 3 1 1

#### Output

3

### Test Case 2

#### Input

11

2 2 3 3 2 2 4 4 3 1 1

#### Output

3

### C++ Program

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int find_odd_occurrence(int arr[], int low, int high)
```

```
{  
    if(low == high)  
    {  
        return low;  
    }  
}
```

```
int mid = (low + high) / 2;
```

```
if(mid & 1)  
{
```

```

    if(arr[mid] == arr[mid - 1])
    {
        return find_odd_occurrence(arr, mid+1, high);
    }
    else
    {
        return find_odd_occurrence(arr, low, mid);
    }
}
else
{
    if(arr[mid] == arr[mid + 1])
    {
        return find_odd_occurrence(arr, mid+2, high);
    }
    else
    {
        return find_odd_occurrence(arr, low, mid);
    }
}
}
int main(void)
{
    int n;
    cin >> n;
    int A[n];
    for(int i = 0; i < n; i++)
    {
        cin >> A[i];
    }
    cout<<A[find_odd_occurrence(A, 0, n-1)];
    return 0;
}

```

## PROGRAM 80

Jarvis is weak in computing palindromes for Alphanumeric characters. While Ironman is busy fighting with Thanos, he needs to activate sonic punch but Jarvis is stuck in computing palindromes. You are given a string S containing alphanumeric characters. Find out whether the string is a palindrome or not. If you are unable to solve it then it may result in the death of Iron Man.

**Input Format:** The first line contains string 'S'.

**Output Format:** The output contains "YES" if the string is palindrome and "NO" if the string is not a palindrome.

**Constraints:**  $1 \leq |S| \leq 100000$

**Note:** Consider alphabets and numbers only for palindrome check. Ignore symbols and white spaces.

### Sample Input:

I am :IronnorI Ma, i

### Sample Output:

YES

### Test Case 1

#### Input

Ab?/Ba

#### Output

Yes

## C Program

```
#include <stdio.h>
#include <string.h>
#define MAX 100000
int is_palindrome(char temp[]);
int main(){
    char str[MAX];
    scanf("%[^\n]s", str);
    char temp[MAX];
    int j = 0, i = 0;
    for(i = 0; str[i] != '\0'; i++){
        if((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= '0' && str[i] <= '9')){
            if(str[i] >= 'A' && str[i] <= 'Z'){
                temp[j++] = str[i] + 32;
            }
            else{
                temp[j++] = str[i];
            }
        }
    }
}
```

```

    }
}
temp[j] = '\0';
if(is_palindrome(temp)){
    printf("Yes");
}
else{
    printf("No");
}
}
int is_palindrome(char temp[]){
    int len = strlen(temp);
    int is_found = 1;
    int s = 0;
    int e = len - 1;
    while(s <= e){
        if(temp[s] != temp[e]){
            is_found = 0;
            break;
        }
        s++;
        e--;
    }
    return is_found;
}

```

### C++ Program

```

#include <iostream>
using namespace std;
#include<string.h>
#define MAX 100000
int is_palindrome(char temp[]);

int main(){
    char str[MAX];
    scanf("%s", str);
    char temp[MAX];
    int j = 0, i = 0;
    for(i = 0; str[i] != '\0'; i++){
        if((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= '0' && str[i] <= '9')){
            if(str[i] >= 'A' && str[i] <= 'Z'){
                temp[j++] = str[i] + 32;
            }
        }
    }
}

```

```

        else{
            temp[j++] = str[i];
        }
    }
}
temp[j] = '\0';
if(is_palindrome(temp)){
    cout<<"Yes";
}
else{
    cout<<"No";
}
}
int is_palindrome(char temp[]){
    int len = strlen(temp);
    int is_found = 1;
    int s = 0;
    int e = len -1;
    while(s <= e){
        if(temp[s] != temp[e]){
            is_found = 0;
            break;
        }
        s++;
        e--;
    }
    return is_found;
}

```

## PROGRAM 81

Given an integer, write a program to swap two bits at given positions in the binary representation of it.

### Input Format:

The first line of input consists of a number n

The second line of input consists of the given positions x and y

**Output Format:** Binary representation of the given number n after swapping the bits.

### Test Case 1

#### Input

31

2 6

#### Output

91

00011111

### Test Case 2

#### Input

50

2 4

#### Output

38

00110010

### Test Case 3

#### Input

32

1 8

#### Output

32

00100000

### C++ Program

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int swap(int n, int p, int q)
```

```
{  
    if(((n & (1 << p)) >> p) ^ ((n & (1 << q)) >> q))  
    {  
        n = n ^ (1 << p);  
        n = n ^ (1 << q);  
    }
```

```
    }  
    return n;  
}  
int main()  
{  
    int n;  
    cin >> n;  
    int x, y;  
    cin >> x >> y;  
  
    cout << swap(n, x, y) << endl << bitset<8>(n);  
    return 0;  
}
```



## PROGRAM 82

Fanny is given a string along with the string which contains single character x. She has to remove the character x from the given string. Help her write a function to remove all occurrences of x character from the given string.

### Input Specification:

Input 1: input string s

Input 2: String containing any character x

**Output Specification:** String without the occurrence of character x

### Example 1:

#### Input:

input1: welcome to mettl

input2: l

#### Output:

welcome to mett

#### Explanation:

As l is the character which is required to be removed, therefore all the occurrences of l are removed, keeping all other characters.

### Example 2:

input1: Lord out of Rings

input2: o

#### Output:

Lrd f Rings

#### Explanation:

As o is the character which is required to be removed, therefore all the occurrences of o are removed, keeping all other characters.

### Test Case 1

#### Input

welcome to mettl

l

#### Output

welcome to mett

### Test Case 2

#### Input

Tamilarasan guna a

#### Output

Tmilrsn gun

## C++ Program

```
#include<iostream>
using namespace std;
string removeCharacter(string str, char ch)
{
    int len = str.length();
    string s = "";
    for(int i = 0; i < len; i++)
    {
        char c = str[i];
        if(c == ch)
            continue;
        else
            s = s + c;
    }
    return s;
}
int main()
{
    string str;
    char ch;
    getline(cin, str);
    cin >> ch;
    cout << removeCharacter(str, ch);
    return 0;
}
```

## PROGRAM 83

### Problem description:

Write a function to find if a given string is palindrome or not. Return 0 if the input string is not a palindrome and 1 if the input string is a palindrome.

**Note:** A string is said to be a palindrome if the reverse of the string is the same as string. For example, “abba” is a palindrome, but “abbc” is not a palindrome.

**Input Specification:** input1: A string of characters

**Output Specification:** 0 or 1 depending on whether the string is a palindrome or not.

### Example 1:

input1: level

Output: 1

Explanation:

Reverse of string “level” is “level”. As, they are same hence the string is a palindrome.

### Example 2:

input1: abcd

Output: 0

Explanation:

Reverse of string “abcd” is “dcba”. As, they are not the same hence the string is not a palindrome.

### Test Case 1

#### Input

level

#### Output

1

### Test Case 2

#### Input

abcd

#### Output

0

### C Program

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

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

```
#include <stdbool.h>
```

```
bool isPalRec(char str[],  
              int s, int e)
```

```

{
    if (s == e)
        return true;
    if (str[s] != str[e])
        return false;
    if (s < e + 1)
        return isPalRec(str, s + 1, e - 1);
    return true;
}

```

```

bool isPalindrome(char str[])
{
    int n = strlen(str);
    if (n == 0)
        return true;

    return isPalRec(str, 0, n - 1);
}

```

```

int main()
{
    char str[15];
    scanf("%s",str);

    if (isPalindrome(str))
        printf("1");
    else
        printf("0");

    return 0;
}

```

### C++ Program

```

#include<bits/stdc++.h>
using namespace std;
void isPalindrome(string str)
{
    int l = 0;
    int h = str.length()- 1;
    while (h > l)
    {
        if (str[l++] != str[h--])
        {

```

```

        cout<<"0";
        return;
    }
}
cout<<"1";
}
int main()
{

    string str;
    cin>>str;
    isPalindrome(str);

    return 0;
}

```

### **JAVA Program**

```

import java.util.*;
class Main
{
    static boolean isPalRec(String str,
                            int s, int e)
    {
        if (s == e)
            return true;

        if ((str.charAt(s)) != (str.charAt(e)))
            return false;
        if (s < e + 1)
            return isPalRec(str, s + 1, e - 1);
        return true;
    }
    static boolean isPalindrome(String str)
    {
        int n = str.length();
        if (n == 0)
            return true;
        return isPalRec(str, 0, n - 1);
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String str =sc.nextLine();
    }
}

```

```
    if (isPalindrome(str))
        System.out.println("1");
    else
        System.out.println("0");
    }
}
```

### **Python**

```
def isPalindrome(s):
    return s == s[::-1]
s = input()
ans = isPalindrome(s)
if ans:
    print("1")
else:
    print("0")
```

## PROGRAM 84

Given a number N, write a program to find the Nth term of the below-mentioned series. 14, 28, 20, 40, 32, 64, 56, 112, 104, 208.....

**Input Format:** The input contains an integer 'N'

**Output Format:** Print the Nth term of the series

**Constraints:**  $1 \leq N \leq 1000$

**Sample Input:**

2

**Sample Output:**

28

**Test Case 1**

**Input**

10

**Output**

208

### C Program

```
#include <stdio.h>
#include<math.h>
int main() {
    int n, b = 14;
    scanf("%d", &n);
    for(int i = 2; i <= n; i++){
        if(i % 2 == 0){
            b = b * 2;
        }
        else{
            b = b - 8;
        }
    }
    printf("%d ", b);
    return 0;
}
```

### C++ Program

```
#include <iostream>
using namespace std;
```

```
#include<math.h>
int main() {
    int n, b = 14;
    cin>>n;
    for(int i = 2; i <= n; i++){
        if(i % 2 == 0){
            b = b * 2;
        }
        else{
            b = b - 8;
        }
    }
    cout<<b;
    return 0;
}
```



## PROGRAM 85

**Problem Description:** The placement season has begun in a college. There are N number of students standing outside an interview room in a line. It is given that a person who goes in first has higher chances of getting selected. Each student has a number associated with them known as the problem-solving capability (PSC). The higher the capability, the higher the chances of selection. Now, each student wants to know the number of students ahead of him/her who have more problem-solving capability than him/her. Find this number for each student.

### Input Specification:

input1: An integer N, which denotes the number of students present.

input2: An array of size N. denoting the problem-solving capability of the students.

### Input:

input1: 6

input2: {4, 9, 5, 3, 2, 10}

**Output:** {0, 0, 1, 3, 4, 0}

### Explanation:

The first student has no one ahead of him/her. So, the answer is 0.

The second student has PSC greater than the first. So, the answer is 0.

Third student: 9 is greater than 5. So, the answer is 1.

Fourth student: 9, 4 and 5 are greater than 3. So, the answer is 3.

Fifth student: 3, 5, 9 and 4 are greater than 2. So, the answer is 4.

Sixth student: No one ahead has a greater PSC. So, the answer is 0.

### Test Case 1

#### Input

6 4 9 5 3 2 10

#### Output

0 0 1 3 4 0

### Test Case 2

#### Input

5 3 4 1 5 2

#### Output

0 0 2 0 3

### C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int n,i,j,count;
    cin>>n;
    int a[n],psc[n];
```

```

for(i=0;i<n;i++)
    cin>>a[i];
for(i=0;i<n;i++)
{
    count=0;
    for(j=0;j<i;j++)
    {
        if(a[i]<a[j])
            count++;
    }
    psc[i]=count;
}
for(i=0;i<n;i++)
    cout<<psc[i]<<" ";
}

```

## PROGRAM 86

Write a program to encipher the given character by using the given key.

In this coding scheme alphabet code of a/A-0, b/B-1, ..., z/Z - 25

### Input format:

The first line input will be a character.

The next line will be a key value (integer type).

**Output format:** Encipher the given character and print the same without framing any extra words

### Sample Input:

B

10

### Sample Output

L

### Explanation:

Add the key value 10 to the B's value i.e.,  $1+10 = 11$  and 11's respective character is L.

### Test Case 1

#### Input

B

10

#### Output

L

### C Program

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

```
int main()
```

```
{
```

```
    char ch,encipher;
```

```
    int key;
```

```
    scanf("%c %d",&ch,&key);
```

```
    // islower(argument) is a function to check whether the character is lower case or not
```

```
    if(islower(ch))
```

```
    {
```

```
        encipher=((ch-97)+key)%26;
```

```
        printf("%c",encipher+97);
```

```
    }
```

```
    else
```

```
    {
```

```
        encipher=((ch-65)+key)%26;
```

```

    printf("%c",encipher+65);
}
return 0;
}

```

### **C++ Program**

```

#include <iostream>
#include<stdio.h>
using namespace std;

int main()
{
    char ch,encipher;
    int key;
    cin>>ch>>key;

    if(islower(ch))
    {
        encipher=((ch-97)+key)%26;

    cout<<(char)(encipher+97);
    }
    else
    {
        encipher=((ch-65)+key)%26;

        cout<<(char)(encipher+65);
    }
    return 0;
}

```

### **JAVA Program**

```

import java.util.*;
class Main
{
    public static void main(String args[])
    {
        char ch;
        int encipher;
        int key;
        Scanner sc = new Scanner(System.in);
        ch = sc.next().charAt(0);
    }
}

```

```

key=sc.nextInt();
if(Character.isLowerCase(ch))
{
    encipher=((int)ch-97)+key)%26;
    System.out.print((char)(encipher+97));
}
else
{
    encipher=((int)ch-65)+key)%26;
    System.out.print((char)(encipher+65));
}
}
}

```

## Python

```

def encrypt(text,s):
    result = ""

    # traverse text
    for i in range(len(text)):
        char = text[i]

        if (char.isupper()):
            result += chr((ord(char) + s-65) % 26 + 65)

        else:
            result += chr((ord(char) + s - 97) % 26 + 97)

    return result

#check the above function
text =input()
s = int(input())
print(encrypt(text,s))

```

## PROGRAM 87

**Problem Description:** Consider the following encoding scheme, A -> 1, B -> 2, ..., Z -> 26

Given an encoding using this scheme (a string containing at most 1000 digits), find the number of possible decodings. The function should return a string containing the number of decodings.

**Note:** The input string contains valide digits from 0 to 9 and ther are no leading 0's, no extra trailing 0's and no two more consecutive 0's.

**Input Specification:** Input: the Input string

**Output Specification:** Return a string representing the number of possible decodings.

### Example 1:

input1: 121

output: 3

### Explanation:

Possible decodings are 'ABA', 'AU' and 'LA'. Hence, number of decodings are 3.

### Example2:

input1: 12345

output: 3

### Explanation:

Possible decodings are 'ABCDE', 'LCDE' and 'AWDE'. Hence, number of decodings are 3.

### Test Case 1

#### Input

121

#### Output

3

### Test Case 2

#### Input

12345

#### Output

3

### C++ Program

```
#include <iostream>
#include <cstring>
using namespace std;
int countDecodingDP(char *digits, int n)
{
    int count[n+1];
```

```

count[0] = 1;
count[1] = 1;
if(digits[0]=='0')
    return 0;
for (int i = 2; i <= n; i++)
{
    count[i] = 0;
    if (digits[i-1] > '0')
        count[i] = count[i-1];
    if (digits[i-2] == '1' ||
        (digits[i-2] == '2' && digits[i-1] < '7') )
        count[i] += count[i-2];
}
return count[n];
}

int main()
{
    char digits[50];
    cin >> digits;
    int n = strlen(digits);
    cout << countDecodingDP(digits, n);
    return 0;
}

```

## PROGRAM 88

Write a C++ program to count the frequency of alphabets in a given string.

**INPUT FORMAT:** Input consists of 1 string.

### SAMPLE INPUT & OUTPUT:

google

e 1

g 2

l 1

o 2

### Test Case 1

#### Input

chrome

#### Output

c 1

e 1

h 1

m 1

o 1

r 1

### Test Case 2

#### Input

Codelab

#### Output

C 1

a 1

b 1

d 1

e 1

l 1

o 1

### Test Case 3

#### Input

Freshgrad

#### Output

F 1

a 1

d 1

e 1

g 1



h 1  
r 2  
s 1

### **C++ Program**

```
#include<iostream>
using namespace std;
#include<string.h>
int main()
{
    char str[100];
    int i;
    cin>>str;
    int freq[256]={0};
    for(i = 0; str[i] != '\0'; i++)
    {
        freq[str[i]]++;
    }
    for(i = 0; i < 256; i++)
    {
        if(freq[i] != 0)
        {
            cout<<(char)int(i)<<" "<<freq[i]<<"\n";
        }
    }
    return 0;
}
```

## PROGRAM 89

Write a program to rearrange the array alternately i.e first element should be a maximum value, the second element should be a minimum value, the third element should be a second max, the fourth element is the second min and so on.

### Input Format:

The first input contains an integer 'n' which denotes the size of the array

The next inputs denote the elements of the array

Note: The input values should not be duplicated.

**Output Format:** Print the rearranged array

### Sample Input:

7  
9 6 5 3 2 7 1

### Sample Output:

9 1 7 2 6 3 5

### Test Case 1

#### Input

5  
4 7 8 9 2

#### Output

9 2 8 4 7

## C Program

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    int arr[n];
    for(int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }
    for(int i = 0; i < n; i++) {
        for(int j = i + 1; j < n; j++) {
            if(i % 2 == 0) {
                if(arr[i] < arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}
```

```

        else if(i % 2 == 1) {
            if(arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}
for(int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
}
}

```

### C++ Program

```
#include <iostream>
```

```
using namespace std;
```

```

int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i = 0; i < n; i++){
        cin>>arr[i];
    }
    for(int i = 0; i < n; i++) {
        for(int j = i + 1; j < n; j++) {
            if(i % 2 == 0) {
                if(arr[i] < arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
            else if(i % 2 == 1) {
                if(arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }
}

```

```
for(int i = 0; i < n; i++) {  
    cout << arr[i] << " ";  
}  
}
```

## PROGRAM 90

Write a C++ program to find whether the given numbers are Friendly Pair or not. Friendly pair is two or more numbers with a common abundancy index i.e the ratio between the sum of divisors of a number and the number itself. For example, 6 and 28 are a Friendly pair as  $(\text{Sum of divisors of } 6)/6 = (\text{Sum of divisors of } 28)/28$  Note: Don't use unnecessary sentences in input or output.

**INPUT & OUTPUT FORMAT:** Input consists of 2 integers. The first integer corresponds to number 1 and the second integer corresponds to number 2. If they are a Friendly pair, display “Friendly Pair” or display “Not Friendly Pair”.

### Test Case 1

#### Input

6  
28

#### Output

Friendly Pair

### Test Case 2

#### Input

13  
12

#### Output

Not Friendly Pair

### Test Case 3

#### Input

28  
6

#### Output

Friendly Pair

## C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int n1,n2;
    int sum1=0,sum2=0;
    cin>>n1>>n2;
    for(int i=1;i<n1;i++)
    {
        if(n1%i==0)
```

```

        sum1=sum1+i;
    }
}
for(int i=1;i<n2;i++)
{
    if(n2%i==0)
    {
        sum2=sum2+i;
    }
}
if((sum1/n1) == (sum2/n2))
{
    cout<<"Friendly Pair";
}
else
{
    cout<<"Not Friendly Pair";
}
return 0;
}

```

## PROGRAM 91

Joffrey Baratheon wanted to excel in Math. He was learning about Kaprekar number from Olenna Tyrell, his Maths teacher. She gave him a few random numbers and asked him to find whether they are Kaprekar number or not. (Consider an n-digit number k. Square it and add the right n digits to the left n or n-1 digits. If the resultant sum is k, then k is called a Kaprekar number. For example, 9 is a Kaprekar number since  $9^2 = 81$  &  $8 + 1 = 9$  297 is a Kaprekar number as  $297^2 = 88209$  &  $88 + 209 = 297$ ) Can you help Joffrey to write a C/C++ program to find whether the given number is a Kaprekar number or not.

**INPUT & OUTPUT FORMAT:** Input consists of a single integer. Refer sample output for details.

### SAMPLE INPUT:

9

### SAMPLE OUTPUT:

Kaprekar Number

### Test Case 1

#### Input

45

#### Output

Kaprekar Number

### Test Case 2

#### Input

3

#### Output

Not a Kaprekar Number

### Test Case 3

#### Input

25

#### Output

Not a Kaprekar Number

### C++ Program

```
#include<iostream>
using namespace std;
#include<math.h>
int main()
{
    int a,c=1;
    cin>>a;
    int n=a;
    while(n/=10)
```

```
c++;  
int sq=a*a;  
int po=pow(10,c);  
int f = sq%po;  
int s=sq/po;  
if(f+s==a)  
    cout<<"Kaprekar Number";  
else  
    cout<<"Not a Kaprekar Number";  
return 0;  
}
```



## PROGRAM 92

A doctor opens up his clinic at 10:00 a.m. There are many patients visiting him for the checkup. The doctor examines only one patient at a time and gives 20 minutes to each patient. There is a waiting room outside the doctor's clinic so that when a patient arrives and the doctor is busy he waits in the waiting room. Patients start arriving from 11:00 a.m. and each new patient arrives at a time period of  $x$  minutes. You are given the total number of patients and at what time interval (in minutes) the next patients are arriving for each test case. Write a program to calculate how much time (in minutes) the last patient needs to wait in the waiting room.

**Input format:** The subsequent inputs denote the total number of patients and time interval (in minutes) in which the next patients are visiting.

**Output format:** You have to calculate the total time in minutes for which the last patient needs to wait in the waiting room.

### Constraints:

$$0 < n < 100$$

$$0 < x < 30$$

### Sample Input:

6 5

### Sample Output:

75

### Test Case 1

#### Input

4 5

#### Output

45

### C Program

```
#include<stdio.h>
int main()
{
    int tpatns, time;
    scanf("%d%d",&tpatns,&time);
    printf("%d",((tpatns-1) * (20 - time)));
    return 0;
}
```

### C++ Program

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int tpatns, time;
    cin>>tpatns>>time;
    cout<<((tpatns-1) * (20 - time));
    return 0;
}
```

### PROGRAM 93

Write a program to generate the first 'n' terms of the following series 0.5, 1.5, 4.5, 13.5, ...

**INPUT & OUTPUT FORMAT:** The input is an integer 'n' which denotes the number of terms to be printed in the series. Print the series and refer the sample output for formatting.

#### SAMPLE INPUT:

5

#### SAMPLE OUTPUT:

0.5 1.5 4.5 13.5 40.5

#### Test Case 1

##### Input

1

##### Output

0.5

#### Test Case 2

##### Input

4

##### Output

0.5 1.5 4.5 13.5

#### Test Case 3

##### Input

6

##### Output

0.5 1.5 4.5 13.5 40.5 121.5

### C++ Program

```
#include <iostream>
```

```
using namespace std;
```

```
#include<math.h>
```

```
int main()
```

```
{  
    int n;  
    cin>>n;  
    float j=0.5;  
    for(int i=1;i<=n;i++)  
    {  
        cout<<j<<" ";
```

```
    j=j*3;  
}  
  
return 0;  
}
```

## PROGRAM 94

You are given a cubic dice with 6 faces. All the individual faces have a number printed on them. The numbers are in the range of 1 to 6, like any ordinary dice. You will be provided with a face of this cube, your task is to guess the number on the opposite face of the cube.

**Input Format:** The first line of the input containing a positive integer N.

**Output Format:** Print the number that is on the opposite side of the given face.

**Constraints:**  $1 \leq N \leq 6$

**Sample Input:**

6

**Sample Output:**

1

**Test Case 1**

**Input**

1

**Output**

6

## C Program

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    switch(n){
        case 1:
            printf("6");
            break;
        case 2:
            printf("5");
            break;
        case 3:
            printf("4");
            break;
        case 4:
            printf("3");
            break;
        case 5:
            printf("2");
            break;
```

```

        case 6:
            printf("1");
            break;
    }
    return 0;
}

```

### **C++ Program**

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin>>n;
    switch(n){
        case 1:
            cout<<"6";
            break;
        case 2:
            cout<<"5";
            break;
        case 3:
            cout<<"4";
            break;
        case 4:
            cout<<"3";
            break;
        case 5:
            cout<<"2";
            break;
        case 6:
            cout<<"1";
            break;
    }
    return 0;
}

```

## PROGRAM 95

Write a program to print the given numbers in words.

**Input Format:** The input contains an integer 'N' which denotes the number

**Output Format:** Print the given number in words

**Constraints:**  $0 \leq N \leq 1000$

### Test Case 1

#### Input

436

#### Output

four hundred thirty six

### C Program

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

```
char* ones[] = {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};
char* two_digit_tens[] = {"", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"};
char* two_digit_special[] = {"ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen",
"eighteen", "nineteen"};
```

```
void print_ones(int n)
```

```
{
    printf("%s ", ones[n]);
}
```

```
void print_double_digit(int n)
```

```
{
    if(n >= 10 && n <= 19)
    {
        printf("%s ", two_digit_special[n - 10]);
    }
    else
    {
        if(n/10 != 0)
        {
            printf("%s ", two_digit_tens[(n/10)]);
        }

```

```
        if(n%10 != 0)
```

```

        {
            print_ones(n%10);
        }
    }
}

void print_triple_digit(int n)
{
    if(n/100 != 0)
    {
        print_ones(n/100);
        printf("hundred ");
    }

    if(n%100 != 0)
    {
        print_double_digit(n%100);
    }
}

void print_four_digit(int n)
{
    if(n/1000 != 0)
    {
        print_ones(n/1000);
        printf("thousand ");
    }

    if(n%1000 != 0)
    {
        print_triple_digit(n%1000);
    }
}

int main()
{
    int n;
    scanf("%d", &n);

    if(n < 10)
    {
        print_ones(n);
    }
    else if(n < 100)

```



```

{
    print_double_digit(n);
}
else if(n < 1000)
{
    print_triple_digit(n);
}
else // n < 9999
{
    print_four_digit(n);
}

return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

```

```

char* ones[] = {"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};
char* two_digit_tens[] = {"", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"};
char* two_digit_special[] = {"ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen",
"eighteen", "nineteen"};

```

```

void print_ones(int n)
{
    printf("%s ", ones[n]);
}

```

```

void print_double_digit(int n)
{
    if(n >= 10 && n <= 19)
    {
        printf("%s ", two_digit_special[n - 10]);
    }
    else
    {
        if(n/10 != 0)
        {
            printf("%s ", two_digit_tens[(n/10)]);
        }

        if(n%10 != 0)

```

```

        {
            print_ones(n%10);
        }
    }
}

void print_triple_digit(int n)
{
    if(n/100 != 0)
    {
        print_ones(n/100);
        cout<<"hundred ";
    }

    if(n%100 != 0)
    {
        print_double_digit(n%100);
    }
}

void print_four_digit(int n)
{
    if(n/1000 != 0)
    {
        print_ones(n/1000);
        cout<<"thousand ";
    }

    if(n%1000 != 0)
    {
        print_triple_digit(n%1000);
    }
}

int main()
{
    int n;
    cin>>n;

    if(n < 10)
    {
        print_ones(n);
    }
    else if(n < 100)

```

```
{  
    print_double_digit(n);  
}  
else if(n < 1000)  
{  
    print_triple_digit(n);  
}  
else // n < 9999  
{  
    print_four_digit(n);  
}  
  
return 0;  
}
```

## PROGRAM 96

The program will receive 3 English words inputs from STDIN

These three words will be read one at a time, in three separate line

The first word should be changed like all vowels should be replaced by \*

The second word should be changed like all consonants should be replaced by @

The third word should be changed like all char should be converted to upper case

Then concatenate the three words and print them

Other than these concatenated words, no other characters/string should or message should be written to STDOUT

For example, if you print how are you then the output should be h\*wa@eYOU.

You can assume that input of each word will not exceed more than 5 chars

### Sample Input:

how

are

you

### Sample Output:

h\*wa@eYOU

### Test Case 1

#### Input

how 999 you

#### Output

h\*w999YOU

## C++ Program

```
#include<iostream>
```

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

```
using namespace std;
```

```
int main()
```

```
{
```

```
int i;
```

```
char a[100],b[100],c[100];
```

```
cin >> a;
```

```
cin >> b;
```

```
cin >> c;
```

```
for(i=0;a[i]!='\0';i++)
```

```
{
```

```
if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u'||a[i]=='A'||a[i]=='E'||a[i]=='I'||a[i]=='O'||a[i]=='U')
```

```
    a[i]='*';
```

```
}
```

```
for(i=0;b[i]!='\0';i++)
```

```
{
```

```
if((b[i]>='a'&&b[i]<='z') || (b[i]>='A'&&b[i]<='Z'))
```

```

        if(!(b[i]=='a'||b[i]=='e'||b[i]=='i'||b[i]=='o'||b[i]=='u'||b[i]=='A'||b[i]=='E'||b[i]=='T'||b[i]=='O'||b[i]=='U'))
            b[i]='@';
    }

    for(i=0;c[i]!='\0';i++)
    {
        if(c[i]>='a'&& c[i]<='z')
            c[i]=c[i]-32;
    }

    cout << a << b << c;
    return 0;
}

```

## PROGRAM 97

1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17.....

This series is a mixture of 2 series. The odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order.

Write a program to find the Nth term in this series.

The value N is a positive integer that should be read as Input. The Nth term that is calculated by the program should be written to STDOUT. Other than the value of Nth term, no other characters / string or message should be written to STDOUT.

**Input:**

14

**Output:**

17

**For example,** when N:14, the 14th term in the series is 17. So only the value 17 should be printed to STDOUT.

### Test Case 1

**Input**

21

**Output**

89

### C++ Program

```
#include<iostream>
#define MAX 99999
using namespace std;
void fibonacci(int n)
{
    int a = 0, b = 1, next;
    for (int i = 1; i<=n; i++)
    {
        next = a + b;
        a = b;
        b = next;
    }
    cout<< a;
}

void prime(int n)
{
    int i, j, flag, count =0;
    for (i=2; i<=MAX; i++)
    {
```

```

    flag = 0;
    for (j=2; j<i; j++)
    {
        if(i%j == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0){
        if(++count == n)
        {
            cout<< i;
            break;
        }
    }
}
}

int main(){
    int n;
    cin >> n;
    if(n%2 == 1)
        fibonacci (n/2 + 1);
    else
        prime(n/2);

    return 0;
}

```

## PROGRAM 98

Wil Wheaton who has filled in the shoes of Professor Proton in the show Professor Proton Fun with Mathematics wanted the kids to learn about Special number. (A 2-digit number is said to be a special number if the sum of sum of its digits and the product of its digits is equal to the number itself. For example, 19 is a special number. The digits in 19 are 1 and 9. The sum of the digits is 10 and the product of the digits is 9.  $10+9 = 19$ .) Can you help Wil to write a C++ program to find all special numbers between 2 limits m and n (both inclusive). Assume that m and n are 2-digit numbers.

**INPUT & OUTPUT FORMAT:** Input consists of 2 integers m and n. Refer to sample output for output format.

### SAMPLE INPUT:

11  
30

### SAMPLE OUTPUT:

19  
29

### Test Case 1

#### Input

12  
45

#### Output

19  
29  
39

### Test Case 2

#### Input

21  
34

#### Output

29

### C++ Program

```
#include<iostream>
using namespace std;
#include<math.h>
int main()
{
    int a,b;
    cin>>a>>b;
    for(int i=a;i<=b;i++)
    {
```



```
if(i<=99)
{
    int sum=i%10+i/10;
    int prod=(i%10)*(i/10);
    if((sum+prod)==i)
        cout<<i<<endl;
}
}
return 0;
}
```

## PROGRAM 99

Amy Farrah Fowler asks her students to give 3 examples for positive odd numbers. When the student specifies a correct answer, his/her score is incremented by 1. When the student specifies a positive even number, his/her score is decremented by 0.5. When the student specifies a negative number, he/she will not be given any more chances to correct his or her mistake and his/her score will be decremented by 1. So a student's turn comes to an end when he/she has correctly specified 3 positive odd numbers or when the student has specified a negative number. Some students didn't know the difference between odd numbers and even numbers and they made many mistakes and so it was difficult for her to maintain the scores. She asks for your help. Can you please help her by writing a program to calculate the score?

**INPUT & OUTPUT FORMAT:** Input consists of a list of integers. The output consists of a single line. The score needs to be corrected to 1 decimal place. [For this exercise, don't worry about duplicate odd numbers. Even if the students specify the same odd number thrice, it is accepted].

### SAMPLE INPUT & OUTPUT 1:

```
1
3
5
3.0
```

### SAMPLE INPUT & OUTPUT 2:

```
1
2
5
6
7
2.0
```

### Test Case 1

#### Input

```
2
-4
```

#### Output

```
-1.5
```

### Test Case 2

#### Input

```
3
3
3
```

#### Output

```
3.0
```

### Test Case 3

**Input**

1  
7  
9

**Output**

3.0

**C++ Program**

```
#include <iostream>
using namespace std;

int main()
{
    int n,count=0;
    float score=0;
    cin>>n;
    while(n>0 && count!=3)
    {
        if(n%2!=0)
        {
            count = count + 1;
            score = score + 1;
        }
        else
        {
            score=score-0.5;
        }
        cin>>n;
    }
    if (n<0)
    {
        score=score-1;
    }
    cout<<score;
}
```

## PROGRAM 100

The Modern World Cinderella, was happy to meet The Prince Charming in the ball room and she danced with him happily. While she was dancing , Prince wanted to check whether Cinderella is the perfect suit for him. He asked her a question to check her knowledge on Arrays. Can you help Cinderella to write a C++ program to find whether the two arrays are the same or not. (Two arrays are said to be the same if the sum of both the arrays is the same and the size of arrays is the same).

**INPUT FORMAT:** Input consists of 2 integers and 2 arrays. Integers correspond to the size of arrays. If two arrays are same, display "Same" else display "Not Same"

### Test Case 1

#### Input

4  
4  
1  
2  
3  
4  
1  
2  
3  
4

#### Output

Same

### Test Case 2

#### Input

3  
3  
1  
2  
1  
2  
3  
6

#### Output

Not Same

### Test Case 3

#### Input

2  
2  
1  
2

3

6

## Output

Not Same

## C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    int m,n,i,j,sa=0,sb=0;
    cin>>m>>n;
    int a[m],b[n];
    for(i=0;i<m;i++)
    {
        cin>>a[i];
        sa+=a[i];
    }
    for(j=0;j<n;j++)
    {
        cin>>b[j];
        sb+=b[j];
    }
    if(m==n)
    {
        if(sa==sb)
        {
            cout<<"Same";
        }
        else
        {
            cout<<"Not Same";
        }
    }
    else
    {
        cout<<"Not Same";
    }
}
```

## PROGRAM 101

Sherlock and John were investigating a murder and the murderer was Ms.Irene Adler. She wanted Sherlock to find her by solving the puzzles given by her and the tasks given by her. Sherlock was able to solve all but one. The last task was to write a C++ program to remove the duplicate elements from an array. Can you help Sherlock?

**INPUT FORMAT:** Input consists of 1 integer and 1 array.

The first integer corresponds to the size of the array.

The next integers correspond to the elements in the array.

**OUTPUT FORMAT:** The output consists of an array with no duplicate elements.

### SAMPLE INPUT:

3  
1  
5  
1

### SAMPLE OUTPUT:

1  
5

### Test Case 1

#### Input

5  
1  
2  
2  
3  
4

#### Output

1  
2  
3  
4

### Test Case 2

#### Input

4  
3  
2  
1  
2

#### Output

3  
2  
1

### **C++ Program**

```
#include<iostream>
using namespace std;
#include<math.h>
int main()
{
    int n,i,j;
    cin>>n;
    int arr[n];
    for(i=0;i<n;i++)
    {
        cin>>arr[i];
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(arr[i]==arr[j])
                break;
        }
        if(i==j)
        {
            cout<<arr[i]<<"\n";
        }
    }
    return 0;
}
```

## PROGRAM 102

Given an unsorted array `Arr[]` and a number `N`. You need to write a program to find if there exists a pair of elements in the array whose difference is `N`.

**Input:** First line of input contains an integer `T` which denotes the number of test cases. Then `T` test cases follow. First line of each test case contains two space separated integers `L` and `N` where `L` denotes the length of array or the number of elements in the array and `N` denotes the difference between two elements. Second line of each test case contains `L` space separated integers which denotes the elements of the array.

**Output:** For each test case, in a new line print `1` if the pair is found otherwise print `-1` if there does not exist any such pair.

### Constraints:

$1 \leq T \leq 100$

$1 \leq L \leq 104$

$1 \leq \text{Arr}[i] \leq 105$

### Input:

2

6 78

5 20 3 2 5 80

5 45

90 70 20 80 50

### Output:

1

-1

### Test Case 1

#### Input

2 5 7 2 5 6 7 9 6 10 10 21 20 31 41 11

#### Output

1

1



## PROGRAM 103

### Sweet seventeen

**Problem statement:** Given a maximum of four digit to the base 17(10 -> A, 11 -> B, 12 -> C, 16 -> G) as input, output its decimal value.

#### Test Case 1

**Input (stdin)**

1A

**Output (stdout)**

27

#### Test Case 2

**Input (stdin)**

23GF

**Output (stdout)**

10980

### C++ Program

```
#include <iostream>
```

```
#include <math.h>
```

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

```
using namespace std;
```

```
int main(){
```

```
    char hex[17];
```

```
    long long decimal;
```

```
    int i = 0, val, len;
```

```
    decimal = 0;
```

```
    cin>> hex;
```

```
    len = strlen(hex);
```

```
    len--;
```

```
    for(i = 0;hex[i]!='\0';i++)
```

```
    {
```

```
        if(hex[i]>='0'&& hex[i]<='9'){
```

```
            val = hex[i] - 48;
```

```
        }
```

```
        else if(hex[i]>='a'&& hex[i]<='g'){
```

```
            val = hex[i] - 97 + 10;
```

```
        }
```

```
        else if(hex[i]>='A'&& hex[i]<='G'){
```

```
            val = hex[i] - 65 + 10;
```

```
        }
```

```
    decimal = decimal + val * pow(17,len);  
    len--;  
}  
  
cout<< decimal;  
  
return 0;  
}
```

## PROGRAM 104

**Word is key** - One programming language has the following keywords that cannot be used as identifiers: break, case, continue, default, defer, else, for, func, goto, if, map, range, return, struct, type, var. Write a program to find if the given word is a keyword or not

### Test Case 1

#### Input (stdin)

defer

#### Output (stdout)

defer is a keyword

### Test Case 2

#### Input (stdin)

while

#### Output (stdout)

while is not a keyword

## C++ Program

```
#include<iostream>
```

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

```
using namespace std;
```

```
int main(){
```

```
    char str[16][10] = {"break", "case", "continue", "default", "defer", "else", "for",  
    "func", "goto", "if", "map", "range", "return", "struct", "type", "var"};
```

```
    char input[20];
```

```
    int flag = 0;
```

```
    cin >> input;
```

```
    for(int i = 0; i<16;i++){
```

```
        if(strcmp(input,str[i]) == 0){
```

```
            flag = 1;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if(flag==1){
```

```
        cout << input << " is a keyword";
```

```
    }
```

```
    else{
```

```
        cout << input << " is not a keyword";
```

```
    }
```

```
    return 0;
```

```
}
```

## PROGRAM 105

### Minting Mints

**Problem statement:** It was one of the places, where people need to get their provisions only through fair price (“ration”) shops. As the elder had domestic and official work to attend to, their wards were asked to buy the items from these shops. Needless to say, there was a long queue of boys and girls. To minimize the tedium of standing in the serpentine queue, the kids were given mints. I went to the last boy in the queue and asked him how many mints he has. He said that the number of mints he has is one less than the sum of all the mints of kids standing before him in the queue. So I went to the penultimate kid to know how many mints she has. She said that if I add all the mints of kids before her and subtract one from it, the result equals the mints she has. It seemed to be the uniform response from everyone. So, I went to the boy in the head of queue consoling myself that he would not give the same response as others. He said, “I have four mints”. Given the number of first kid’s mints (n) and the length (len) of queue as input, write a program to display the total number of mints with all the kids.

**Constraints:**  $2 < n < 10$ ;  $1 < \text{len} < 20$

#### Test Case 1

**Input (stdin)**

4 2

**Output (stdout)**

7

#### Test Case 2

**Input (stdin)**

14 4

**Output (stdout)**

105

### C++ Program

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int s,n,sum=0;
    cin>>s>>n;
    int a[n];
    a[0]=sum=s;
    for(int i=1;i<n;i++)
    {
        a[i]=sum-1;
        sum=sum+a[i];
    }
    cout<<sum<<" ";
}
```

## PROGRAM 106

### To zero or not to zero

**Problem statement:** Given a pair of positive integers m and n ( $m < n$ ;  $0 < m < 999$ ;  $1 < n \leq 999$ ), write a program to smartly affix zeroes, while printing the numbers from m to n.

#### Example-1

##### Input

5 10

##### Expected output

05 06 07 08 09 10

#### Example-2

##### Input

9 100

##### Expected output

009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034  
035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060  
061 062 063 064 065 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087  
088 089 090 091 092 093 094 095 096 097 098 099 100

#### Case 1

##### Input (stdin)

1 9

##### Output (stdout)

1 2 3 4 5 6 7 8 9

#### Case 2

##### Input (stdin)

3 7

##### Output (stdout)

3 4 5 6 7

### C++ Program

```
#include <iostream>
using namespace std;
int main()
{
    int up,low;
    cin >> low >> up;
    for(int i=low; i<=up; i++)
    {
        if(up>=100)
            printf("%03d ",i);
```

```
else if(up>=10)
    printf("%02d ",i);
else
    printf("%d ",i);
}
```

## PROGRAM 107

Our hoary culture had several great persons since time immemorial and king vikramaditya's nava ratnas (nine gems) belongs to this ilk. They are named in the following shloka:

धनवंतरी क्षषणकाडमरसिंह राडू चेठालमदृ धटकपर्परः कर्मिदाक  
ख्यति कराहमिहिरि नृमृते समाभ्यम रत्नति वै कस्मस्मिनति तिम्दम्

Among these, Varahamihira was an astrologer of eminence and his book Brihat Jataak is recokened as the ultimate authority in astrology.

He was once talking with Amarasimha, another gem among the nava ratnas and the author of Sanskrit thesaurus, Amarakosha.

Amarasimha wanted to know the final position of a person, who starts from the origin 0 0 and travels per following scheme.

He first turns and travels 10 units of distance

His second turn is upward for 20 units

Third turn is to the left for 30 units

Fourth turn is the downward for 40 units

Fifth turn is to the right(again) for 50 units

... And thus he travels, every time increasing the travel distance by 10 units.

**Constraints:**  $2 \leq n \leq 1000$

### Test Case 1

**Input (stdin)**

3

**Output (stdout)**

-20 20

### Test Case 2

**Input (stdin)**

4

**Output (stdout)**

-20 -20

### C++ Program

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n;
```

```

cin>>n;
char c = 'R';
int x = 0, y = 0;
while(n){
    switch(c){
        case 'R':
            x = abs(x) + 10;
            y = abs(y);
            c = 'U';
            break;
        case 'U':
            y = y + 20;
            c = 'L';
            break;
        case 'L':
            x = -(x + 10);
            c = 'D';
            break;
        case 'D':
            y = -(y);
            c = 'R';
            break;
    }
    n--;
}
cout<< x<< " " << y;
}

```



## PROGRAM 108

**Oddly even** - Given a maximum of 100 digit numbers as input, find the difference between the sum of odd and even position digits.

### Input 1:

4567

### Expected output:

2

### Explanation

Sum of odd position digits 4 and 6 is 10. Sum of even position digits 5 and 7 is 12. The difference is  $12-10=2$ .

### Input 2:

5476

### Expected output:

2

### Test Case 1

#### Input (stdin)

4567

#### Output (stdout)

2

### Test Case 2

#### Input (stdin)

9834698765123

#### Output (stdout)

1

## C Program

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

```
int main()
{
    int a = 0, b = 0, i = 0, n;
    char num[100];
    scanf("%s", num);
    n = strlen(num);
    while(n > 0)
    {
        if(i == 0)
        {
```

```

        a+=num[n-1]-48;
        n--;
        i=1;
    }
    else
    {
        b+=num[n-1]-48;
        n--;
        i=0;
    }
}
printf("%d",abs(a-b));

return 0;
}

```

### C++ Program

```

#include <iostream>
#include <string.h>
#include <stdlib.h>

using namespace std;

int main()
{
    int a = 0,b = 0,i = 0, n;
    char num[100];

    cin>> num;
    n = strlen(num);
    while(n>0)
    {
        if(i==0)
        {
            a+=num[n-1]-48;
            n--;
            i=1;
        }
        else
        {
            b+=num[n-1]-48;
            n--;
            i=0;
        }
    }
}

```

```

    }
}
cout<< abs(a-b);

return 0;
}

```

### **JAVA Program**

```

import java.util.*;
public class Main
{
    public static void main(String[] args) {
        Scanner sin = new Scanner(System.in);
        String s=sin.nextLine();
        long num = 0, num1 = 0;
        num=num + s.charAt(0)-'0';
        for(int i=1;i<s.length();i++)
        {
            if(i%2==0)
                num = num + s.charAt(i)-'0';
            else
                num1 = num1 + s.charAt(i)-'0';
        }
        System.out.println(Math.abs(num-num1));
    }
}

```

### **Python Program**

```

num = [int(d) for d in str(input())]
even,odd = 0,0
for i in range(0,len(num)):
    if i % 2 ==0:
        even = even + num[i]
    else:
        odd = odd + num[i]

print(abs(odd-even))

```

## PROGRAM 109

**Geometric Series** - Consider the following series: 1,1,2,3,4,9,8,27,16,81,32,243,64,729,128,2187... Write a program to find the Nth term in the series. This series is a mixture of 2 series - all the odd terms in this series form a geometric series and all the even terms form yet another geometric series. The value N is a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than value of nth term, no other character / string or message should be written to STDOUT.

### Test Case 1

**Input (stdin)**

5

**Output (stdout)**

4

### Test Case 2

**Input (stdin)**

10

**Output (stdout)**

81

### Test Case 3

**Input (stdin)**

11

**Output (stdout)**

32

## C Program

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    if(n % 2 == 1)
    {
        int a = 1;
        int r = 2;
        int term_in_series = (n+1)/2;
        int res = a * pow(2, term_in_series - 1);
        printf("%d ", res);
    }
    else
    {
        int a = 1;
        int r = 3;
```

```

    int term_in_series = n/2;
    int res = a * pow(3, term_in_series - 1);
    printf("%d ", res);
}
return 0;
}

```

### C++ Program

```

#include<iostream>
#include <math.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if(n % 2 == 1)
    {
        int a = 1;
        int r = 2;
        int terms= (n+1)/2;
        int res = a * pow(2, terms - 1);
        cout<<res;
    }
    else
    {
        int a = 1;
        int r = 3;
        int terms = n/2;
        int res = a * pow(3, terms - 1);
        cout<<res;
    }
    return 0;
}

```

## PROGRAM 110

**ODD/EVEN Series** - Consider the below series : 0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8 Write a program to find the nth term in this series. This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous term using the formula  $(x/2)$  . The value n is a positive integer that should be read from STDIN and the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.

### Test Case 1

**Input (stdin)**

5

**Output (stdout)**

4

### Test Case 2

**Input (stdin)**

10

**Output (stdout)**

4

### Test Case 3

**Input (stdin)**

100

**Output (stdout)**

49

## C Program

```
#include<stdio.h>
int main()
{
    int n,a,d,t_s1,t_s2,n_term;
    scanf("%d",&n);
    if(n%2==1)
    {
        a=0,d=2;
        t_s1=(n+1)/2;
        n_term=a+(t_s1-1)*d;
        printf("%d",n_term);
    }
    else
    {
        a=0,d=1;
        t_s2=n/2;
```

```

        n_term=a+(t_s2-1)*d;
        printf("%d",n_term);
    }
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main()
{
    int n,a,d,t_s1,t_s2,n_term;
    cin>>n;
    if(n%2==1)
    {
        a=0,d=2;
        t_s1=(n+1)/2;
        n_term=a+(t_s1-1)*d;
        cout<<n_term;
    }
    else
    {
        a=0,d=1;
        t_s2=n/2;
        n_term=a+(t_s2-1)*d;
        cout<<n_term;
    }

    return 0;
}

```

## PROGRAM 111

**Sum of Previous terms** - Consider the series given below: 1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,..... Write a program to find the Nth term in this series.

This series is formed as below:

1.term(1)=1

2.term(2)=2

3.term(N)=term(N-1)+term(N-2)for N>2

The value N is a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT, other than the value of nth term no other characters /strings and messages should be written to STDOUT.

For example if N =15, the value of 15th n term is 987 which is the sum of 13th and 14th terms .

You can assume that the value of n will not exceed 30

### Test Case 1

**Input (stdin)**

15

**Output (stdout)**

987

### Test Case 2

**Input (stdin)**

10

**Output (stdout)**

89

### Test Case 3

**Input (stdin)**

100

**Output (stdout)**

Invalid Input

## C Program

```
#include<stdio.h>
int main()
{
    int n, t1=1,t2=2,cur=0,i;
    scanf("%d",&n);
    if(n>30)
    {
        printf("Invalid Input");
        return 0;
    }
    else
```



```

{
    if(n==1 || n==2)
    {
        printf("%d",n);
    }
    else
    {
        for(i=3;i<=n;i++)
        {
            cur=t1+t2;
            t1=t2;
            t2=cur;
        }
        printf("%d",cur);
    }
}
}

```

### C++ Program

```

#include<iostream>
using namespace std;
int main()
{
    int n, t1=1,t2=2,cur=0,i;
    cin>>n;
    if(n>30)
    {
        cout<<"Invalid Input";
        return 0;
    }
    else
    {
        if(n==1 || n==2)
        {
            cout<<n;
        }
        else
        {
            for(i=3;i<=n;i++)
            {
                cur=t1+t2;
                t1=t2;
                t2=cur;
            }
        }
    }
}

```

```
    }  
    cout<<cur;  
    }  
}  
return 0;  
}
```

## PROGRAM 112

**Encipher it !!!** - Write a program to encipher the given character by using the given key. In this coding scheme alphabet code of a/A-0, b/B-1, ..., z/Z - 25

### Input format:

The first line input will be a character.

The next line will be a key value (integer type).

**Output format:** Encipher the given character and print the same without framing any extra words

### Sample Input:

B  
10

### Sample Output

L

**Explanation:** Add the key value 10 to the B's value i.e.,  $1+10 = 11$  and 11's respective character is L.

### Case 1

#### Input (stdin)

z  
1

#### Output (stdout)

a

## C Program

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

```
int main()
```

```
{
```

```
    char ch,encipher;
```

```
    int key;
```

```
    scanf("%c %d",&ch,&key);
```

```
    // islower(argument) is a function to check whether the character is lower case or not
```

```
    if(islower(ch))
```

```
    {
```

```
        encipher=((ch-97)+key)%26;
```

```
        printf("%c",encipher+97);
```

```
    }
```

```
    else
```

```
    {
```

```
        encipher=((ch-65)+key)%26;
```

```
        printf("%c",encipher+65);
```

```
    }  
    return 0;  
}
```

### **C++ Program**

```
#include <iostream>  
#include<stdio.h>  
using namespace std;
```

```
int main()  
{  
    char ch,encipher;  
    int key;  
    cin>>ch>>key;  
  
    if(islower(ch))  
    {  
        encipher=((ch-97)+key)%26;  
  
    cout<<(char)(encipher+97);  
    }  
    else  
    {  
        encipher=((ch-65)+key)%26;  
  
        cout<<(char)(encipher+65);  
    }  
    return 0;  
}
```

## PROGRAM 113

**Character count in given String** - Given an Input string aaaabbccdee. Count the no of same consecutive character and give the output as a4b2c2d1e2. Assume: Maximum Length of the string is 20. Based on the above example, write a program that accept the input string through STDIN and writes the transformed string to STDOUT. Other than the transformed string, no other character / string / message should be written to STDOUT.

### Test Case 1

#### Input (stdin)

aaaabbccdee

#### Output (stdout)

a4b2c2d1e2

### Test Case 2

#### Input (stdin)

AAAZZZZZCCBA

#### Output (stdout)

A3Z5C2B1A1

### Test Case 3

#### Input (stdin)

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

#### Output (stdout)

Invalid Input

## C Program

```
#include<stdio.h>
#include<stdlib.h>
int main(int c,char **arg)
{
    char a[30];
    scanf("%s",a);
    int count=1,i,len=0;
    for(len=0;a[len]!='\0';len++);
    if(len>20)
    {
        printf("Invalid Input");
        return 0;
    }
    for(i=1;i<=len;i++)
    {
        if(a[i]==a[i-1])
        {
            count++;
        }
    }
}
```

```

    }
    else
    {
        printf("%c%d",a[i-1],count);
        count=1;
    }
}
}

```

### C++ Program

```

#include<stdio.h>
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string a;
    cin>>a;

    int count=1,i,len=a.length();
    //cout<<len;
    if(len>20)
    {
        cout<<"Invalid Input";
        return 0;
    }
    else
    {
        for(i=1;i<=len;i++)
        {
            if(a[i]==a[i-1])
            {
                count++;
            }
            else
            {
                cout<<a[i-1]<<count;
                count=1;
            }
        }
    }
    return 0;
}

```

## PROGRAM 114

**Swap the character** - Given three strings as an input. Change the vowels of the first string to \$. Change the consonants of the second string to #. Convert the entire third string from lowercase to uppercase. Finally, concatenate all these three strings and print the output.

### Test Case 1

#### Input (stdin)

AIR

FORCE

aCaDeMy

#### Output (stdout)

\$\$R#O##EACADEMY

### C Program

```
#include<stdio.h>
#include<stdlib.h>
char isVowel(char c,int n)
{
    if(((c=='A' || c=='E' || c=='I' || c=='O' || c=='U' || c=='a' || c=='e' || c=='i' || c=='o' || c=='u') && n==1)
    {
        return '$';
    }
    else if((c!='A' && c!='E' && c!='I' && c!='O' && c!='U' && c!='a' && c!='e' && c!='i' && c!='o' && c!='u') && n==2)
    {
        return '#';
    }
    else if((c>='a' && c<='z') && n==3)
    {
        return c-32;
    }
    return c;
}
int main()
{
    int i,j;
    char string[3][100];
    for(i=0;i<3;i++)
    {
        scanf("%s",string[i]);
    }
    for(i=0;i<3;i++)
    {
        for(j=0;string[i][j]!='\0';j++)
```

```

    {
        printf("%c",isVowel(string[i][j],i+1));
    }
}
}

```

### C++ Program

```

#include <iostream>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
char isVowel(char c,int n)
{
    if((c=='A' || c=='E' || c=='I' || c=='O' || c=='U' || c=='a' || c=='e' || c=='i' || c=='o' || c=='u') && n==1)
    {
        return '$';
    }
    else if((c!='A' && c!='E' && c!='I' && c!='O' && c!='U' && c!='a' && c!='e' && c!='i' && c!='o' && c!='u') && n==2)
    {
        return '#';
    }
    else if((c>='a' && c<='z') && n==3)
    {
        return c-32;
    }
    return c;
}
int main()
{
    int i,j;
    char string[3][100];
    for(i=0;i<3;i++)
    {
        scanf("%s",string[i]);
    }
    for(i=0;i<3;i++)
    {
        for(j=0;string[i][j]!='\0';j++)
        {
            cout<<isVowel(string[i][j],i+1);
        }
    }
}

```



## PROGRAM 115

**Remove duplicates and reverse** - Write a program that will replace multiple consecutive occurrences of a character with a single occurrence, and print the result in the reverse order. The input stream of characters should be read from STDIN and the result should be written to STDOUT. Other than the required output, no other characters / string / message should be written to STDOUT. You can assume that the input string length will not exceed 30 characters.

### Test Case 1

#### Input (stdin)

Ball

#### Output (stdout)

laB

### Test Case 2

#### Input (stdin)

Hello

#### Output (stdout)

oleH

### Test Case 3

#### Input (stdin)

HeLloh

#### Output (stdout)

holLeH

## C Program

```
#include<stdio.h>
int main()
{
    char str[30], temp[30];
    int i, count=0;
    scanf("%[^\\n]s", str);
    for(i = 0; str[i]!='\\0'; i++)
    {
        if(str[i] != str[i+1])
        {
            temp[count] = str[i];
            count++;
        }
    }
    for(i = count-1; i>=0; i--)
    {
        printf("%c", temp[i]);
    }
}
```

```
}  
return 0;  
}
```

### **C++ Program**

```
#include <iostream>  
#include<stdio.h>  
using namespace std;  
int main()  
{  
    char str[30], temp[30];  
    int i, count=0;  
    scanf("%[^\\n]s", str);  
    for(i = 0; str[i]!='\\0'; i++)  
    {  
        if(str[i] != str[i+1])  
        {  
            temp[count] = str[i];  
            count++;  
        }  
    }  
    for(i = count-1; i>=0; i--)  
    {  
        cout<<temp[i];  
    }  
    return 0;  
}
```

## PROGRAM 116

**Change the case of alphabet** - Write a program to change the case of the given alphabet and print

### Test Case 1

**Input (stdin)**

a

**Output (stdout)**

A

### Test Case 2

**Input (stdin)**

A

**Output (stdout)**

a

### C Program

```
#include <stdio.h>
int main() {
    char c;
    scanf("%c", &c);

    // Upper to lower case
    if('A' <= c && c <= 'Z')
    {
        printf("%c",c+32);
    }

    // Lower to upper case
    if('a' <= c && c <= 'z')
    {
        printf("%c",c-32);
    }

    return 0;
}
```

### C++ Program

```
#include <iostream>
#include <stdio.h>
using namespace std;

int main()
```

```
{  
    char c;  
    cin>>c;  
    if('A' <= c && c <= 'Z')  
        printf("%c",c+32);  
    if('a' <= c && c <= 'z')  
        printf("%c",c-32);  
    return 0;  
}
```

## PROGRAM 117

**Area of a circle** - Program to find the area of a circle. The input diameter will be given as an integer, the output area should be printed as a floating point value with 2 point precision. No other extra information should be printed except the area value to the stdout. (Assume  $\text{PI} = 3.14$ )

### Test Case 1

#### Input (stdin)

6

#### Output (stdout)

28.26

### Test Case 2

#### Input (stdin)

20

#### Output (stdout)

314

### C Program

```
#include<stdio.h>
int main()
{
    int dia;
    float r, area;
    scanf("%d",&dia);
    r=(float)dia/2;
    area=3.14*r*r;
    printf("%0.2f",area);
    return 0;
}
```

### C++ Program

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
    int dia;
    float r, area;
    cin>>dia;
    r=(float)dia/2;
    area=3.14*r*r;
    printf("%0.2f",area);
    return 0;
}
```

## PROGRAM 118

**Hypotenuse of a triangle** - Program to find the hypotenuse of a triangle. Get the opposite and adjacent sides from the user and calculate and display the hypotenuse of the given triangle, The output is a floating point value with precision 2.

### Test Case 1

#### Input (stdin)

2.5

3.5

#### Output (stdout)

4.30

### Test Case 2

#### Input (stdin)

5.8

6.8

#### Output (stdout)

8.94

### Test Case 3

#### Input (stdin)

1

6

#### Output (stdout)

6.08

## C Program

```
#include <stdio.h>
#include<math.h>
int main()
{
    float hyp, opp, adj;
    scanf("%f%f", &opp, &adj);
    hyp=sqrt((opp*opp) + (adj*adj));
    printf("%.2f", hyp);
    return 0;
}
```

## C++ Program

```
#include <stdio.h>
#include <iostream>
#include<math.h>
```

```
using namespace std;
int main()
{
    float hyp, opp, adj;
    cin>>opp>>adj;
    hyp=sqrt((opp*opp) + (adj*adj));
    printf("%.2f", hyp);
    return 0;
}
```

## PROGRAM 119

**Factorial** - Find the factorial value for the given number. Factorial for n value will be  $1*2*3*...n$

### Test Case 1

**Input (stdin)**

5

**Output (stdout)**

120

### Test Case 2

**Input (stdin)**

7

**Output (stdout)**

5040

### Test Case 3

**Input (stdin)**

12

**Output (stdout)**

479001600

### C Program

```
#include<stdio.h>
int main()
{
    long int n,f=1;
    scanf("%ld",&n);
    for(int i=1;i<=n;i++)
    {
        f=f*i;
    }
    printf("%ld",f);
}
```

### C++ Program

```
//#include<stdio.h>
#include <iostream>
using namespace std;
int main()
{
    long int n,f=1;
    cin>>n;
```



```
for(int i=1;i<=n;i++)  
{  
    f=f*i;  
}  
cout<<f;  
}
```

## PROGRAM 120

**Length of string** - Program to find the length of string. Get a string input and print its length.

### Test Case 1

#### Input (stdin)

Hello world

#### Output (stdout)

11

### Test Case 2

#### Input (stdin)

My project

#### Output (stdout)

11

### C Program

```
#include <stdio.h>
int main()
{
    char s[1000], i;
    scanf("%[^\\n]s", s);
    for(i = 0; s[i] != '\\0'; ++i);
    printf("%d", i);
    return 0;
}
```

### C++ Program

```
#include<iostream>
#include <stdio.h>
using namespace std;
int main()
{
    char s[1000], i;
    scanf("%[^\\n]s", s);
    for(i = 0; s[i] != '\\0'; ++i);
    printf("%d", i);
    return 0;
}
```

## PROGRAM 121

**Perfect number** - Check whether a number is perfect number or not. A perfect number is a positive integer that is equal to the sum of its proper positive divisors.

### Example:

#### Input:

6

#### Output:

Yes ( 1 + 2 + 3 = 6 )

### Test Case 1

#### Input (stdin)

7

#### Output (stdout)

No

### C Program

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

### C++ Program

```
#include <iostream>
#include <cctype>
using namespace std;

int main(){
    int n,i=1,sum=0;
    cin >> n;
```

```
    while(i<n){  
        if(n%i==0)  
            sum=sum+i;  
        i++;  
    }  
    if(sum==n)  
        cout <<"Yes";  
    else  
        cout <<"No";  
  
    return 0;  
}
```

## PROGRAM 122

**GCD** - Write a program to calculate the Greatest common factor of 2 numbers. Use using math skills to compute GCD of given 2 numbers. Input will consist of 2 lines each containing 1 number each

### Test Case 1

**Input (stdin)**

6  
3

**Output (stdout)**

3

### Test Case 2

**Input (stdin)**

4  
5

**Output (stdout)**

1

### Test Case 3

**Input (stdin)**

6  
2

**Output (stdout)**

2

## C Program

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

```
int main()
{
    int a, b;
    int c;
    scanf("%d", &a);
    scanf("%d", &b);
    while ( a != 0 ) {
        c = a; a = b%a; b = c;
    }
    printf("%d\n", b);
    return 0;
}
```

## C++ Program

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a, b, c;
```

```
    cin>>a>>b;
```

```
    while ( a != 0 )
```

```
    {
```

```
        c = a;
```

```
        a = b%a;
```

```
        b = c;
```

```
    }
```

```
    cout<<b;
```

```
    return 0;
```

```
}
```

## PROGRAM 123

**Decimal to binary** - Write a program to convert decimal number to binary. Convert a given decimal integer number n to its binary equivalent.

### Test Case 1

**Input (stdin)**

3

**Output (stdout)**

11

### Test Case 2

**Input (stdin)**

100

**Output (stdout)**

1100100

### Test Case 3

**Input (stdin)**

456

**Output (stdout)**

111001000

## C Program

```
#include<stdio.h>
#include<stdlib.h>
int main(int a, char* argv[])
{
    int n, cnt,i;
    int b[32];
    scanf("%d",&n);
    cnt=0;
    while(n!=0)
    {
        b[cnt]=n%2;
        n=n/2;
        cnt++;
    }
    for(i=(cnt-1);i>=0;i--)
    printf("%d",b[i]);
    return 0;
}
```

## C++ Program

```
#include<iostream>
#include<stdlib.h>
using namespace std;

int main()
{
    int n, cnt=0,i;
    int b[32];
    cin>>n;
    while(n!=0)
    {
        b[cnt]=n%2;
        n=n/2;
        cnt++;
    }
    for(i=(cnt-1);i>=0;i--)
        cout<<b[i];
    return 0;
}
```



## PROGRAM 124

**Simple Interest** - Write a program to calculate the simple interest. Input consists of 3 values. 1 - Principle (P) 2 - No.of. years (n) 3 - Rate of Interest (r) The output consists of one value Simple Interest (SI)

### Sample Input:

1200  
2  
5.4

### Sample output:

Simple Interest = 129.600006

### Test Case 1

#### Input (stdin)

10000  
4  
20

#### Output (stdout)

8000.000000

### Test Case 2

#### Input (stdin)

100.12  
7  
1.2

#### Output (stdout)

8.410081

## C Program

```
#include<stdio.h>
int main()
{
    float P,n,r,SI;
    scanf("%f%f%f",&P,&n,&r);
    SI = (P*n*r)/100;
    printf("%f",SI);
}
```

## C++ Program

```
#include<iostream>
#include<stdio.h>
using namespace std;
```

```
int main()
{
    float P,n,r,SI;
    cin>>P>>n>>r;
    SI = (P*n*r)/100;
    printf("%f",SI);
    return 0;
}
```

## PROGRAM 125

**Sum of Prime Numbers** - Write a program to find the sum of all prime numbers in a given range. The program should consider all the prime numbers within the range, excluding the upper bound and lower bound.

### Test Case 1

#### Input (stdin)

7  
24

#### Output (stdout)

83

### Test Case 2

#### Input (stdin)

0  
12

#### Output (stdout)

28

## C Program

```
#include <stdio.h>

int main()
{
    int i, j, end, start, isPrime, sum=0;
    scanf("%d%d",&start,&end);
    for(i=start+1; i<end; i++)
    {
        isPrime = 1;
        for(j=2; j<=i/2 ;j++)
        {
            if(i%j==0)
            {
                isPrime = 0;
                break;
            }
        }
        if(isPrime==1)
        {
            sum += i;
        }
    }
    printf("%d",sum);
    return 0;
}
```

## C++ Program

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, sum=0, i, j, flag;

    cin >> a; // Take input

    cin >> b; // Take input

    for (i = a; i <= b; i++) {
        if (i == 1 || i == 0 || i==a || i==b)
            continue;
        flag = 1;

        for (j = 2; j <= i / 2; ++j) {
            if (i % j == 0) {
                flag = 0;
                break;
            }
        }
        if (flag == 1)
            sum=sum+i;
    }
    cout<< sum;

    return 0;
}
```

## PROGRAM 126

**Perfect square** - Program to check whether the integer is a perfect square. Get an input from the user and display "YES" if it is a perfect square else display " NO" as output

### Test Case 1

**Input (stdin)**

25

**Output (stdout)**

YES

### Test Case 2

**Input (stdin)**

55

**Output (stdout)**

NO

### Test Case 3

**Input (stdin)**

4

**Output (stdout)**

YES

## C Program

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

## C++ Program

```
#include <iostream>
#include<math.h>
using namespace std;

bool isPerfectSquare(long double x)
{
    long double sr = sqrt(x);
    return ((sr - floor(sr)) == 0);
}

int main() {
    long double x;
    cin>>x;
    if (isPerfectSquare(x))
        cout << "YES";
    else
        cout << "NO";
    return 0;
}
```

## PROGRAM 127

**Centigrade to Fahrenheit.** - Write a program to input temperature in Centigrade and convert to Fahrenheit.

### Test Case 1

#### Input (stdin)

100

#### Output (stdout)

212.00

### Test Case 2

#### Input (stdin)

54

#### Output (stdout)

129.20

### Test Case 3

#### Input (stdin)

-100

#### Output (stdout)

-148.00

### C Program

```
#include<stdio.h>
int main()
{
    float celsius, fahrenheit;
    scanf("%f", &celsius);
    fahrenheit = (celsius * 9 / 5) + 32;
    printf("%.2f", fahrenheit);
    return 0;
}
```

### C++ Program

```
#include <iostream>
#include<stdio.h>
using namespace std;

int main()
{
    float celsius, fahrenheit;
    cin>>celsius;
    fahrenheit = (celsius * 9 / 5) + 32;
```

```
printf("%0.2f", fahrenheit);  
return 0;  
}
```



## PROGRAM 128

**Fibonacci series** - Program to print the Fibonacci series up to n number of terms. Get the n from user and display the fibonacci series upto the n elements

### Test Case 1

**Input (stdin)**

10

**Output (stdout)**

0 1 1 2 3 5 8 13 21 34

### Test Case 2

**Input (stdin)**

4

**Output (stdout)**

0 1 1 2

### Test Case 3

**Input (stdin)**

20

**Output (stdout)**

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

## C Program

```
#include <stdio.h>
int main()
{
    int i, n, t1 = 0, t2 = 1, next;
    scanf("%d", &n);
    for (i = 1; i <= n; ++i)
    {
        printf("%d ", t1);
        next = t1 + t2;
        t1 = t2;
        t2 = next;
    }
    return 0;
}
```

## C++ Program

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int n, t1 = 0, t2 = 1, nextTerm = 0;
    cin >> n;
    for (int i = 1; i <= n; ++i)
    {
        if(i == 1)
        {
            cout << t1;
            continue;
        }
        if(i == 2)
        {
            cout << " "<<t2<<" ";
            continue;
        }
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;

        cout << nextTerm << " ";
    }
    return 0;
}

```

## PROGRAM 129

**Nth Fibonacci Number** - Write a program to find the nth Fibonacci number. Given a number n, your program should output the nth number in the Fibonacci sequence.

### Test Case 1

**Input (stdin)**

3

**Output (stdout)**

1

### Test Case 2

**Input (stdin)**

6

**Output (stdout)**

5

### Test Case 3

**Input (stdin)**

8

**Output (stdout)**

13

## C Program

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

```
int main()
{
    int n,t1=1,t2=0,next,i;
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        next=t1+t2;
        t2=t1;
        t1=next;
    }
    printf("%d",t2);
}
```

## C++ Program

```
#include<bits/stdc++.h>
using namespace std;
```

```
int fib(int n)
{
    if (n <= 1)
        return n;
    return fib(n-1) + fib(n-2);
}
```

```
int main ()
{
    int n;
    cin>>n;
    cout << fib(n-1);
    getchar();
    return 0;
}
```

## PROGRAM 130

**Binary to Octal** - Write a program to convert the given binary number to its equivalent octal value.

**Input Format:** Input consists of integer

**Output Format:** Refer the sample output format

### Test Case 1

**Input (stdin)**

10101

**Output (stdout)**

25

### Test Case 2

**Input (stdin)**

101010011

**Output (stdout)**

523

## C Program

```
#include<stdio.h>
int power_fn(int base,int exp)
{
    int i,sum=1;
    for(i=0;i<exp;i++)
    {
        sum=sum*base;
    }
    return sum;
}
int main()
{
    int binary,sum=0,rem,exp=0;
    scanf("%d",&binary);
    while(binary!=0)
    {
        rem=binary%10;
        sum=sum+rem*power_fn(2,exp);
        exp++;
        binary=binary/10;
    }
    int array[100],i=0,j;
    while(sum!=0)
```

```

{
    array[i++]=sum%8;
    sum=sum/8;
}
for(j=i-1;j>=0;j--)
    printf("%d",array[j]);
}

```

### C++ Program

```

#include<iostream>
#include<stdio.h>
using namespace std;

int power_fn(int base,int exp)
{
    int i,sum=1;
    for(i=0;i<exp;i++)
    {
        sum=sum*base;
    }
    return sum;
}

int main()
{
    int binary,sum=0,rem,exp=0;
cin>>binary;
    while(binary!=0)
    {
        rem=binary%10;
        sum=sum+rem*power_fn(2,exp);
        exp++;
        binary=binary/10;
    }
    int array[100],i=0,j;
    while(sum!=0)
    {
        array[i++]=sum%8;
        sum=sum/8;
    }
    for(j=i-1;j>=0;j--)
    cout<<array[j];
}

```

## PROGRAM 131

**Reverse a given number** - Program to reverse a given number. Get an integer from the user and print the reverse of that number.

### Test Case 1

**Input (stdin)**

2324324

**Output (stdout)**

4234232

### Test Case 2

**Input (stdin)**

9754

**Output (stdout)**

4579

### Test Case 3

**Input (stdin)**

789520

**Output (stdout)**

25987

## C Program

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

## C++ Program

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n, reversedNumber = 0, remainder;
    cin >> n;

    while(n != 0)
    {
        remainder = n%10;
        reversedNumber = reversedNumber*10 + remainder;
        n /= 10;
    }

    cout << reversedNumber;

    return 0;
}
```



## PROGRAM 132

**String palindrome** - Write a Program to check whether a string palindrome or not.

### Sample input:

wow

### Sample output:

wow is a palindrome

### Test Case 1

#### Input (stdin)

apple

#### Output (stdout)

apple is not a palindrome

### Test Case 2

#### Input (stdin)

madam

#### Output (stdout)

madam is a palindrome

## C Program

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

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

```
int main(){
    char string1[20];
    int i, length;
    int flag = 0;

    scanf("%s", string1);

    length = strlen(string1);

    for(i=0;i < length ;i++){
        if(string1[i] != string1[length-i-1]){
            flag = 1;
            break;
        }
    }

    if (flag) {
        printf("%s is not a palindrome", string1);
    }
}
```

```

else {
    printf("%s is a palindrome", string1);
}
return 0;
}

```

### **C++ Program**

```

#include <iostream>
#include<string.h>

```

```

using namespace std;

```

```

int main(){
    char str1[20];
    int i, length;
    int flag = 0;
    cin >> str1;

    length = strlen(str1);

    for(i=0;i < length ;i++){
        if(str1[i] != str1[length-i-1]){
            flag = 1;
            break;
        }
    }

    if (flag) {
        cout << str1 << " is not a palindrome" << endl;
    }
    else {
        cout << str1 << " is a palindrome" << endl;
    }
    return 0;
}

```

## PROGRAM 133

**Decimal to Octal** - Write a program to convert decimal to octal.

### Test Case 1

**Input (stdin)**

3

**Output (stdout)**

3

### Test Case 2

**Input (stdin)**

76

**Output (stdout)**

114

### Test Case 3

**Input (stdin)**

21

**Output (stdout)**

25

## C Program

```
#include <stdio.h>
#include <math.h>
int convertDecimalToOctal(int decimalNumber);
int main()
{
    int decimalNumber;
    //printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);
    printf("%d",convertDecimalToOctal(decimalNumber));
    return 0;
}
int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i = 1;
    while (decimalNumber != 0)
    {
        octalNumber += (decimalNumber % 8) * i;
        decimalNumber /= 8; i *= 10;
    }
    return octalNumber;
}
```

## C++ Program

```
#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std;

int convertDecimalToOctal(int decimalNumber);
int main()
{
    int decimalNumber;
    cin>>decimalNumber;
    cout<<convertDecimalToOctal(decimalNumber);
    return 0;
}

int convertDecimalToOctal(int decimalNumber)
{
    int octalNumber = 0, i = 1;
    while (decimalNumber != 0)
    {
        octalNumber += (decimalNumber % 8) * i;
        decimalNumber /= 8; i *= 10;
    }
    return octalNumber;
}
```

## PROGRAM 134

**Greatest of Three Number** - Write a program to print the greatest of given three numbers.

### Test Case 1

#### Input (stdin)

7  
8  
9

#### Output (stdout)

9

### Test Case 2

#### Input (stdin)

100  
1000  
-1000

#### Output (stdout)

1000

### Test Case 3

#### Input (stdin)

0  
1  
-1

#### Output (stdout)

1

## C Program

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

```
int main()
```

```
{  
    int n1,n2,n3;  
    scanf("%d,%d,%d", &n1, &n2, &n3);  
    if((n1 > n2) && (n1 > n3))  
    {  
        printf("%d", n1);  
    }  
    else if(n2 > n3)  
    {  
        printf("%d", n2);  
    }  
    else
```

```

{
    printf("%d", n3);
}
return 0;
}

```

### **C++ Program**

```

#include <iostream>
#include<stdio.h>
using namespace std;
int main()
{
    int n1, n2, n3;
    scanf("%d,%d,%d", &n1, &n2, &n3);
    if(n1 > n2 && n1 > n3)
    {
        cout<< n1;
    }
    else if(n2 > n3)
    {
        cout<< n2;
    }
    else
        cout << n3;
    return 0;
}

```

### **JAVA Program**

```

import java.util.*;
public class Main
{
    public static void main(String[] args) {
        int a,b,c;
        Scanner sc = new Scanner(System.in);
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        if(a > b && a > c)
            System.out.println( a);
        else if(b > a && b > c)
            System.out.println(b);
        else

```

```
System.out.println(c);  
}  
}
```

### **Python Program**

```
a = int(input())  
b = int(input())  
c = int(input())  
if(a > b and a > c):  
    print(a)  
elif(b > c and b > c):  
    print(b)  
else:  
    print(c)
```

## PROGRAM 135

**Second greatest of three numbers** - Write a program to find the second greatest of three numbers.

### Sample Input:

5  
7  
8

### Sample Output:

7

### Test Case 1

#### Input (stdin)

1 2 3

#### Output (stdout)

2

### Test Case 2

#### Input (stdin)

9 8 10

#### Output (stdout)

9

## C Program

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

```
int main()
```

```
{
```

```
    int n1, n2, n3;
```

```
    scanf("%d%d%d",&n1,&n2,&n3);
```

```
    if(n1 > n2 && n1 > n3)
```

```
    {
```

```
        if(n2 > n3)
```

```
        {
```

```
            printf("%d", n2);
```

```
        }
```

```
    else
```

```
    {
```

```
        printf("%d", n3);
```

```
    }
```

```
}
```

```
else if (n2 > n3)
```

```
{
```

```
    if(n1 > n3)
```



```

{
printf("%d", n1);
}
else
{
printf("%d", n3);
}
}
else
{
if(n1 > n2)
{
printf("%d", n1);
}
else
{
printf("%d", n2);
}
}
return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

```

```

int main()
{
int n1, n2, n3;
cin>>n1>> n2>> n3;
if(n1 > n2 && n1 > n3)
{
if(n2 > n3)
{

cout<<n2;
}
else
{
cout<<n3;
}
}
else if (n2 > n3)

```

```
{
    if(n1 > n3)
    {
        cout<<n1;
    }
    else
    {
        cout<<n3;
    }
}
else
{
    if(n1 > n2)
    {
        cout<<n1;
    }
    else
    {cout<<n2;
    }
}
return 0;
}
```

## PROGRAM 136

**Circumference of a circle** - Write a program to find the circumference of a circle. The input radius must be a float variable, the output circumference should be printed as a floating point value with 2 point precision. No other extra information should be printed except the circumference value to the stdout. (Assume  $\text{PI} = 3.14$ )

### Test Case 1

**Input (stdin)**

5

**Output (stdout)**

31.40

### Test Case 2

**Input (stdin)**

7

**Output (stdout)**

43.96

### C Program

```
#include<stdio.h>
int main()
{
    float r,c;
    scanf("%f",&r);
    c=2*3.14*r;
    printf("%0.2f",c);
    return 0;
}
```

### C++ Program

```
#include <iostream>
#include<stdio.h>
using namespace std;
int main()
{
    float r,c;
    cin>>r;
    c=2*3.14*r;
    printf("%0.2f",c);
    return 0;
}
```

## PROGRAM 137

**Roots of a quadratic equation** - Write a program to find the roots of a given quadratic equation. The equation will be in the form of  $ax^2 + bx + c = 0$ . The input will be 3 integers a, b and c and the output will be the roots of the equation. The roots need to be floating point integers with 2 precision digits.

### Output format example:

root1 = -1.00 root2 = -6.00

#### Case 1

##### Input (stdin)

1  
-1  
-6

##### Output (stdout)

root1 = 3.00 root2 = -2.00

#### Case 2

##### Input (stdin)

1  
7  
6

##### Output (stdout)

root1 = -1.00 root2 = -6.00

#### Case 3

##### Input (stdin)

1  
-4  
4

##### Output (stdout)

root1 = 2.00 root2 = 2.00

## C Program

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a, b, c, determinant, r1, r2, real, img;
    scanf("%f %f %f",&a, &b, &c);
    determinant = b*b - 4*a*c;

    if (determinant > 0)
    {
```

```

r1 = (-b + sqrt(determinant)) / (2 * a);
r2 = (-b - sqrt(determinant)) / (2 * a);
printf("root1 = %.2f root2 = %.2f", r1 , r2);
}
else if (determinant == 0)
{
r1 = r2 = -b / (2*a);
printf("root1 = %.2f root2 = %.2f", r1 , r2);
}
else
{
real = -b/(2*a);
img = sqrt(-determinant) / (2*a);
printf("root1 = %.2f + %.2fi root2 = %.2f - %.2fi", real, img, real, img);
}
return 0;
}

```

### C++ Program

```

#include <iostream>
#include <cmath>
#include<stdio.h>
using namespace std;

int main() {

float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
cin >> a >> b >> c;
discriminant = b*b - 4*a*c;

if (discriminant > 0) {
x1 = (-b + sqrt(discriminant)) / (2*a);
x2 = (-b - sqrt(discriminant)) / (2*a);
printf("root1 = %.2f ",x1);
printf("root2 = %.2f",x2);
}

else if (discriminant == 0) {
x1 = (-b + sqrt(discriminant)) / (2*a);
printf("root1 = %.2f ",x1);
printf("root2 = %.2f ",x1);
}
}

```

```
else {  
    realPart = -b/(2*a);  
    imaginaryPart =sqrt(-discriminant)/(2*a);  
    printf("root1 = %.2f + %.2fi ",realPart,imaginaryPart);  
    printf("root2 = %.2f - %.2fi",realPart,imaginaryPart);  
}  
  
return 0;  
}
```

## PROGRAM 138

**Binary to decimal** - Program to convert Binary to decimal. Convert the binary to equivalent decimals, the input is in binary and the output should be decimal.

### Test Case 1

**Input (stdin)**

1010101011

**Output (stdout)**

683

### C Program

```
#include <stdio.h>

void main()
{
    int num, binary_val, decimal_val = 0, base = 1, rem;
    scanf("%d", &num); /* maximum five digits */
    binary_val = num;
    while (num > 0)
    {
        rem = num % 10;
        decimal_val = decimal_val + rem * base;
        num = num / 10 ;
        base = base * 2;
    }

    printf("%d", decimal_val);
}
```

### C++ Program

```
#include<iostream>
#include <stdio.h>
using namespace std;

int main()
{
    int num, bin, dec = 0, base = 1, rem;
    cin>>num; /* maximum five digits */
    bin = num;
    while (num > 0)
    {
        rem = num % 10;
```

```
    dec = dec + rem * base;  
    num = num / 10 ;  
    base = base * 2;  
}  
  
cout<<dec;  
return 0;  
}
```



## PROGRAM 139

**Digital Sum** - Write a program to find the sum of the given digit until a single digit is obtained.

**For example:**

12345 => 6

### Test Case 1

**Input (stdin)**

12345

**Output (stdout)**

6

### Test Case 2

**Input (stdin)**

10123

**Output (stdout)**

7

### C Program

```
#include<stdio.h>
int sumofdigits(int num);
int main()
{
    int num, sod;
    scanf("%d", &num);
    sod = num;
    while(sod >9)
    {
        sod = sumofdigits(sod);
    }
    printf("%d", sod);
    return 0;
}
int sumofdigits(int num)
{
    int rem, sum=0;
    while(num != 0)
    {
        rem = num % 10;
        sum = sum + rem;
        num = num / 10;
    }
    return sum;
}
```

## C++ Program

```
#include<iostream>
using namespace std;

int sumofdigits(int num);
int main()
{
    int num, sod;
    cin>>num;
    sod = num;
    while(sod >9)
    {
        sod = sumofdigits(sod);
    }
    cout<<sod;
    return 0;
}
int sumofdigits(int num)
{
    int rem, sum=0;
    while(num != 0)
    {
        rem = num % 10;
        sum = sum + rem;
        num = num / 10;
    }
    return sum;
}
```

## PROGRAM 140

**Reverse Fibonacci** - Write a program to print the Fibonacci series in the reverse order for the given input.

### Test Case 1

**Input (stdin)**

5

**Output (stdout)**

3 2 1 1 0

### Test Case 2

**Input (stdin)**

10

**Output (stdout)**

34 21 13 8 5 3 2 1 1 0

### C Program

```
#include<stdio.h>
void rev_fib(int arr[], int num);
int main()
{
    int num, arr[100];
    scanf("%d", &num);
    reversefibonacci(arr, num);
}
void reversefibonacci(int arr[], int num)
{
    int i;
    arr[0] = 0;
    arr[1] = 1;
    for(i=2; i<num; i++)
    {
        arr[i] = arr[i-2]+arr[i-1];
    }
    for(i=num-1; i>=0; i--)
    {
        printf("%d ", arr[i]);
    }
}
```

### C++ Program

```
#include<iostream>
using namespace std;
```

```

void revfib(int arr[], int num)
{
    int i;
    arr[0] = 0;
    arr[1] = 1;
    for(i=2; i<num; i++)
    {
        arr[i] = arr[i-2]+arr[i-1];
    }
    for(i=num-1; i>=0; i--)
    {
        cout<<arr[i]<<" ";
    }
}

int main()
{
    int num, arr[100];
    cin>>num;
    revfib(arr, num);
    return 0;
}

```

## PROGRAM 141

**Remove vowels** - Program for removal of vowels from a given sentence. Get a string as sentence and print the sentence without vowels.

### Test Case 1

#### Input (stdin)

Hai Hello Welcome

#### Output (stdout)

H Hll Wlcm

### Test Case 2

#### Input (stdin)

Interactive platform

#### Output (stdout)

ntrectv pltrm

## C Program

```
#include <stdio.h>
#include <string.h>
int check_vowel(char);
int main()
{
    char s[100], t[100];
    int i, j = 0;
    scanf("%[^\\n]s", s);
    for(i = 0; s[i] != '\\0'; i++) {
        if(check_vowel(s[i]) == 0) {
            t[j] = s[i];
            j++;
        }
    }
    t[j] = '\\0';
    strcpy(s, t);
    printf("%s\\n", s);
    return 0;
}
int check_vowel(char c)
{
    switch(c) {
        case 'a':
        case 'A':
        case 'e':
        case 'E':
```

```

    case 'i':
    case 'I':
    case 'o':
    case 'O':
    case 'u':
    case 'U':
        return 1;
    default:
        return 0;
}
}

```

### C++ Program

```

#include <iostream>
#include <stdio.h>
#include <string.h>
#include <cstring>
using namespace std;
int vowelChk(char);
int main(){
    char s[50], t[50];
    int c, d = 0;
    scanf("%s",s) ;
    for(c = 0; s[c] != '\0'; c++) {
        if(vowelChk(s[c]) == 0){
            t[d] = s[c];
            d++;
        }
    }
    t[d] = '\0';
    strcpy(s, t);
    cout<<s;
    return 0;
}
int vowelChk(char ch){
    if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O' || ch == 'u' || ch == 'U')
        return 1;
    else
        return 0;
}

```

## PROGRAM 142

**Smallest of four Numbers** - Write a program to find the Smallest of given four numbers

### Test Case 1

#### Input (stdin)

1 2 3 4

#### Output (stdout)

1

### Test Case 2

#### Input (stdin)

9 8 7 6

#### Output (stdout)

6

### C Program

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

## C++ Program

```
#include<iostream>
using namespace std;

int main()
{
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    if(a<b&&a<c&&a<d)
        cout<<a;
    else if(b<c&&b<d)
        cout<<b;
    else if(c<d)
        cout <<c;
    else
        cout<<d;
    return 0;
}
```



## PROGRAM 143

**Arc length of a circle** - Program to find the arc length of a circle. The input radius and center angle must be a float variables, the output should also be printed as a floating point value with 2 point precision. No other extra information should be printed except the arc length value to the stdout. (Assume  $\text{PI} = 3.14$ )

### Test Case 1

#### Input (stdin)

25

50

#### Output (stdout)

21.81

### Test Case 2

#### Input (stdin)

72.56

36.24

#### Output (stdout)

45.87

### Test Case 3

#### Input (stdin)

5

180

#### Output (stdout)

15.70

### C Program

```
#include<stdio.h>
int main()
{
    float r,angle,len;
    scanf("%f%f",&r,&angle);
    len=2*3.14*r*(angle/360);
    printf("%.2f",len);
    return 0;
}
```

### C++ Program

```
#include<iostream>
#include <stdio.h>
using namespace std;
```

```
int main()
{
float r,angle,len;
cin>>r>>angle;
len=2*3.14*r*(angle/360);
printf("%.2f",len);
return 0;
}
```

## PROGRAM 144

**Linear search** - Write a program to search an element using Linear search. Implements linear search (Searching algorithm) which is used to find whether a given number is present in an array and if it is present then at what location it occurs.

### Test Case 1

**Input (stdin)**

6

11 15 6 8 9 1

6

**Output (stdout)**

3

### Test Case 2

**Input (stdin)**

6

11 15 26 38 9 10

83

**Output (stdout)**

83 isn't present in the array.

### Test Case 3

**Input (stdin)**

7

2 4 -2 7 8 3 9

-2

**Output (stdout)**

3

## C Program

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

```
int main()
```

```
{
```

```
    int array[100], e1, i, n;
```

```
    scanf("%d", &n);
```

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

```
    {
```

```
        scanf("%d", &array[i]);
```

```
    }
```

```

scanf("%d", &e1);

int isFound = 0, e1_index;
for (i = 0; i < n; i++)
{
    if (array[i] == e1)
    {
        e1_index = i;
        isFound = 1;
        break;
    }
}

if(isFound == 1)
{
    printf("%d", e1_index + 1);
}
else
{
    printf("%d isn't present in the array.\n", e1);
}

return 0;
}

```

### C++ Program

```

#include <iostream>
using namespace std;

int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main(void)
{
    int arr[50], x, n;
    cin >> n;
    for(int i=0; i<n; i++)

```

```
cin>>arr[i];
cin>>x;
int result = search(arr, n, x);
(result == -1)? cout<<x<<" isn't present in the array.":cout<<result+1;
return 0;
}
```

## PROGRAM 145

**Power of a Number** - Find the power of a number. Get base and exponent from the user. Power of a Number. Print "Wrong input" if the value of the exponent is negative.

### Test Case 1

#### Input (stdin)

3

4

#### Output (stdout)

81

### Test Case 2

#### Input (stdin)

-3

3

#### Output (stdout)

-27

## C Program

```
#include <stdio.h>
int main()
{
    int base,exp,ans=1,i;
    scanf("%d%d", &base,&exp);
    if(exp>=0)
    {
        for(i=1;i<=exp;i++)
        {
            ans=ans*base;
        }
        printf("%d",ans);
    }
    else
    {
        printf("Wrong input");
    }
}
```

## C++ Program

```
#include <iostream>
using namespace std;
int main()
```

```

{
    int base, exponent;
    long long result = 1;
    cin>>base;
    cin>>exponent;
    if(base!=0 && exponent>=0)
    {
        while (exponent != 0)
        {
            result *= base;
            --exponent;
        }
        cout<<result;
    }
    else
        cout<<"Wrong input";
    return 0;
}

```

### **JAVA Program**

```

import java.util.*;
import java.util.Scanner;
public class Main
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        long result = 1;
        int base=s.nextInt();
        int exponent=s.nextInt();
        if(base!=0 && exponent>=0)
        {
            while (exponent != 0)
            {
                result *= base;
                --exponent;
            }
            System.out.print(result);
        }
        else
            System.out.print("Wrong input");
    }
}

```

## PROGRAM 146

**Find 2 numbers in a given array** - Write a program to find two numbers and print their index else print -1.

### Test Case 1

#### Input (stdin)

5  
3  
6  
7  
8  
9  
7  
9

#### Output (stdout)

Element 1 index = 2  
Element 2 index = 4

### Test Case 2

#### Input (stdin)

7  
1 2 3 4 5 6 7  
5 9

#### Output (stdout)

Element 1 index = 4  
Element 2 index = -1

### Test Case 3

#### Input (stdin)

12  
100 121 131 140 25 22 26 190 -98 -123 -90 -100  
27  
21

#### Output (stdout)

Element 1 index = -1  
Element 2 index = -1

### C Program

```
#include<stdio.h>
int main()
{
    int n,i,num1,num2,num1_index=-1,num2_index=-1;
    scanf("%d",&n);
    int a[n];
```



```

for(i=0;i<n;i++)
{
    scanf("%d",&a[i]);
}
scanf("%d%d",&num1,&num2);
for(i=0;i<n;i++)
{
    if(num1==a[i] && num1_index==-1)
    {
        num1_index=i;
    }
    if(num2==a[i] && num2_index==-1)
    {
        num2_index=i;
    }
}
printf("Element 1 index = %d\nElement 2 index = %d",num1_index,num2_index);
}

```

### C++ Program

```

#include<iostream>
#include<stdio.h>

using namespace std;

int main()
{
    int n,i,num1,num2,num1_index=-1,num2_index=-1;
    cin>>n;
    int a[n];
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    cin>>num1>>num2;
    for(i=0;i<n;i++)
    {
        if(num1==a[i] && num1_index==-1)
        {
            num1_index=i;
        }
        if(num2==a[i] && num2_index==-1)
        {

```

```
        num2_index=i;
    }
}
cout<<"Element 1 index = "<<num1_index<<"\nElement 2 index = "<<num2_index;
}
```

## PROGRAM 147

**Printing unique elements in an array** - Write a program to print all the unique elements in the given array.

**Input Format:** The first line of input denotes the size of an array(N) The remaining line of input denotes the 'N' elements.

**Output Format:** Print all the unique elements

**Sample Input:**

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

**Sample Output:**

```
3
```

### Case 1

**Input (stdin)**

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

**Output (stdout)**

```
2 3 5 6
```

### Case 2

**Input (stdin)**

```
5
11
12
12
31
42
```

**Output (stdout)**

```
11 31 42
```

## C Program

```
#include <stdio.h>

int main()
{
    int n,i,j,flag=0;
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n;i++)
    {
        flag=0;
        for(j=0;j<n;j++)
        {
            if(arr[i]==arr[j] && i!=j)
            {
                flag=1;
                break;
            }
        }
        if(flag==0)
        {
            printf("%d ",arr[i]);
        }
    }
    return 0;
}
```

## C++ Program

```
#include<iostream>
#include <stdio.h>
using namespace std;

int main()
{
    int n,i,j,flag=0;
    cin>>n;
    int arr[n];
    for(i=0;i<n;i++)
    {
        cin>>arr[i];
    }
}
```

```
}  
for(i=0;i<n;i++)  
{  
    flag=0;  
    for(j=0;j<n;j++)  
    {  
        if(arr[i]==arr[j] && i!=j)  
        {  
            flag=1;  
            break;  
        }  
    }  
    if(flag==0)  
    {  
        cout<<arr[i]<<" ";  
    }  
}  
return 0;  
}
```

## PROGRAM 148

**Perfect couple** - Write a program to find pairs of elements whose sum is equal to the given value.

### Sample Input:

5  
1 3 4 7 5  
7

### Sample Output:

Perfect couple: 3 4

### Test Case 1

#### Input (stdin)

6  
1 8 9 11 0 2  
5

#### Output (stdout)

No perfect couple found!

### Test Case 2

#### Input (stdin)

7  
1 4 5 3 11 0 2  
16

#### Output (stdout)

Perfect couple: 5 11

## C Program

```
#include<stdio.h>
#include<stdlib.h>
void check_sum_and_display(int arr[], int size, int sum);
int main()
{
    // Get the size of an array
    int size;
    scanf("%d", &size);

    // Get the array elements
    int arr[50], i;
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
}
```

```

// Get the sum value (to check with an array elements)
int sum;
scanf("%d", &sum);

// Function call to check the sum of any two elements in an array equal to given sum
// and display the same
check_sum_and_display(arr, size, sum);
return 0;
}

void check_sum_and_display(int arr[], int size, int sum)
{
    int i,j;
    for(i=0; i<size-1; i++)
    {
        for(j=i+1; j<size; j++)
        {
            if(sum == (arr[i] + arr[j]))
            {
                printf("Perfect couple: %d %d", arr[i], arr[j]);
                exit(0);
            }
        }
    }
    printf("No perfect couple found!");
}

```

### C++ Program

```

#include<iostream>
#include<stdio.h>
#include<stdlib.h>
using namespace std;

void check_sum_and_display(int arr[], int size, int sum);
int main()
{
    int size;
    cin>>size;

    int arr[50], i;
    for(i=0; i<size; i++)
    {
        cin>>arr[i];
    }
}

```

```

    }

    int sum;
    cin>>sum;

    check_sum_and_display(arr, size, sum);
    return 0;
}

void check_sum_and_display(int arr[], int size, int sum)
{
    int i,j;
    for(i=0; i<size-1; i++)
    {
        for(j=i+1; j<size; j++)
        {
            if(sum == (arr[i] + arr[j]))
            {
                cout<<"Perfect couple: "<<arr[i]<<" "<<arr[j];
                exit(0);
            }
        }
    }
    cout<<"No perfect couple found!";
}

```



## PROGRAM 149

**Vowel or consonant** - Write a program to check whether the given character is vowel or a consonant.

### Test Case 1

**Input (stdin)**

a

**Output (stdout)**

Vowel

### Test Case 2

**Input (stdin)**

D

**Output (stdout)**

Consonant

### Test Case 3

**Input (stdin)**

E

**Output (stdout)**

Vowel

## C Program

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

```
int main()
```

```
{
```

```
    char c;
```

```
    scanf("%c",&c);
```

```
    if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||  
       c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
```

```
    {
```

```
        printf("Vowel", c);
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("Consonant", c);
```

```
    }
```

```
    return 0;
```

```
}
```

## C++ Program

```
#include <iostream>
using namespace std;
```

```
int main()
{
    char c;
    cin>>c;

    if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
       c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
    {
        cout<<"Vowel";
    }
    else
    {
        cout<<"Consonant";
    }
    return 0;
}
```

## PROGRAM 150

**Matrix Addition** - Write a program to add 2 matrices. Input consists of multiple test cases. For each case, the first line contains two integers m1, n1 the size of matrix and m1 following rows containing the integers for each row. It is followed by data for matrix B. For each test case output a single line containing all elements of the resultant matrix.

### Test Case 1

#### Input (stdin)

```
1
2 3
1 2 3
4 5 6
1 2 3
4 5 6
```

#### Output (stdout)

```
2 4 6 8 10 12
```

### Test Case 2

#### Input (stdin)

```
1
1 2
0 1
1 2
```

#### Output (stdout)

```
1 3
```

### Test Case 3

#### Input (stdin)

```
1
1 1
1
1
```

#### Output (stdout)

```
2
```

## C Program

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

```
#define MAX_SIZE 10
```

```
int main()
```

```
{
    int t, sum;
```

```

int first[MAX_SIZE][MAX_SIZE];
int second[MAX_SIZE][MAX_SIZE];
int mul[MAX_SIZE][MAX_SIZE];
int i, j, k, m, n, l;
scanf("%d", &t);

for(l = 0; l < t; l++)
{
    scanf("%d %d", &m, &n);
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            scanf("%d", &first[i][j]);
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            scanf("%d", &second[i][j]);

    for ( i = 0 ; i < m ; i++ )
    {
        for ( j = 0 ; j < n ; j++ )
        {
            mul[i][j] = first[i][j] + second[i][j];
        }
    }
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            printf("%d ", mul[i][j]);
    printf("\n");
}
return 0;
}

```

### C++ Program

```

#include<iostream>
#include<stdio.h>
#define MAX_SIZE 10
using namespace std;

```

```

int main()
{
    int t, sum;
    int first[10][10];
    int second[10][10];
    int mul[10][10];

```

```

int i, j, k, m, n, l;
cin>>t;

for(l = 0; l < t; l++)
{
    cin>>m>>n;
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            cin>>first[i][j];
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            cin>>second[i][j];

    for ( i = 0 ; i < m ; i++ )
    {
        for ( j = 0 ; j < n ; j++ )
        {
            mul[i][j] = first[i][j] + second[i][j];
        }
    }
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            cout<<mul[i][j]<<" ";
    cout<<"\n";
}
return 0;
}

```

## PROGRAM 151

**Highest frequency Character** - Write a program to find the highest frequency character in a string.

### Test Case 1

#### Input (stdin)

me and my friends love programming

#### Output (stdout)

Maximum occurring character is 'M' = 4 times.

### Test Case 2

#### Input (stdin)

Navy Academy for Career Enhancement

#### Output (stdout)

Maximum occurring character is 'E' = 6 times.

### Test Case 3

#### Input (stdin)

Try! Try! Try! until you succeed

#### Output (stdout)

Maximum occurring character is 'T' = 4 times.

## C Program

```
#include<stdio.h>
int main()
{
    char str[100];
    int ascii[26]={ },i,max=0,max_char;
    scanf("%[^\n]s",str);
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]>=97 && str[i]<=122)
            str[i]=str[i]-32;
        if(str[i]!=' ')
        {
            ascii[str[i]-65]++;
        }
    }
    for(i=0;i<26;i++)
    {
        if(ascii[i]>max)
        {
            max=ascii[i];
            max_char=i+65;
        }
    }
}
```

```

    }
}
printf("Maximum occurring character is '%c' = %d times.", max_char,max);
}

```

### C++ Program

```

#include<iostream>
#include<stdio.h>
using namespace std;

int main()
{
    char str[100];
    int ascii[26]={ },i,max=0,max_char;
    scanf("%s",str);
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]>=97 && str[i]<=122)
            str[i]=str[i]-32;
        if(str[i]!=' ')
        {
            ascii[str[i]-65]++;
        }
    }
    for(i=0;i<26;i++)
    {
        if(ascii[i]>max)
        {
            max=ascii[i];
            max_char=i+65;
        }
    }
    char c=(char)max_char;
    cout<<"Maximum occurring character is "<<c<<" = "<<max<<" "<<"times.";
}

```

## PROGRAM 152

**Count of words in a given string** - Write a program to print the number of words in a given string

### Test Case 1

#### Input (stdin)

Hello world

#### Output (stdout)

2

### Test Case 2

#### Input (stdin)

Interactive programming environment

#### Output (stdout)

3

### Test Case 3

#### Input (stdin)

Army academy for career enhancement

#### Output (stdout)

5

### Test Case 4

#### Input (stdin)

Welcome 4399 to programming

#### Output (stdout)

4

## C Program

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



```
    printf("%d",word);
}
```

### **C++ Program**

```
#include<iostream>
#include <stdio.h>
using namespace std;

int main()
{
    char str[100];
    int i,word=0;
    scanf("%[^\\n]s",str);
    for(i=0;str[i]!='\\0';i++)
    {
        if(str[i]==' ')
        {
            word++;
        }
    }
    word++;
    cout<<word;
}
```

### **JAVA Program**

```
import java.util.Scanner;
class Main
{
    public static void main(String args[])
    {
        String text;
        int countWords = 0;

        Scanner SC=new Scanner(System.in);
        text=SC.nextLine();

        // word count
        for(int i=0; i<text.length()-1; i++)
        {
            if(text.charAt(i)==' ')
                countWords++;
        }
    }
}
```

```
        System.out.println(countWords+1);  
    }  
}
```

## PROGRAM 153

**Maximum element in an array** - Write a program to find the maximum element in an array.

### Test Case 1

#### Input (stdin)

5  
1 3 2 5 4

#### Output (stdout)

5

### Test Case 2

#### Input (stdin)

7  
-1 -2 -3 3 2 1 -100

#### Output (stdout)

3

### Test Case 3

#### Input (stdin)

2  
1 -1

#### Output (stdout)

1

### C Program

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

## C++ Program

```
#include <iostream>
using namespace std;

int main()
{
    int array[100], maximum, size, c, location = 1;
    cin>>size;
    for (c = 0; c < size; c++)
        cin>>array[c];
    maximum = array[0];
    for (c = 1; c < size; c++)
    {
        if (array[c] > maximum)
        {
            maximum = array[c];
            location = c+1;
        }
    }
    cout<<maximum;
    return 0;
}
```

## PROGRAM 154

**Segregate 0s and 1s** - Write a program to segregate 0s and 1s. You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array.

### Test Case 1

**Input (stdin)**

6  
0 1 0 1 1 1

**Output (stdout)**

0 0 1 1 1 1

### Test Case 2

**Input (stdin)**

10  
0 1 0 1 0 0 1 1 1 0

**Output (stdout)**

0 0 0 0 0 1 1 1 1 1

## C Program

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

## C++ Program

```
#include<iostream>
using namespace std;

void segregate0and1(int arr[], int n)
{
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == 0)
            count++;
    }
    for (int i = 0; i < count; i++)
        arr[i] = 0;

    for (int i = count; i < n; i++)
        arr[i] = 1;
}

void print(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

int main()
{
    int arr[100],n,i;
    cin>>n;
    for(i=0;i<n;i++)
        cin>>arr[i];
    segregate0and1(arr, n);
    print(arr, n);

    return 0;
}
```

## **PROGRAM 155**

**Printing Duplicate Elements** - Write a program to print all the duplicate elements in an array.

### **Test Case 1**

#### **Input (stdin)**

6

1 2 3 4 2 4 5

#### **Output (stdout)**

2 4

### **Test Case 2**

#### **Input (stdin)**

10

1 2 3 20 10 15 10 3 2 1

#### **Output (stdout)**

1 2 3 10

### **Test Case 3**

#### **Input (stdin)**

10

10 20 30 20 25 5 15 5 9 12

#### **Output (stdout)**

20 5

## PROGRAM 156

Given a sorted array `arr[]` of size `N` without duplicates, and given a value `x`. Find the floor of `x` in given array. Floor of `x` is defined as the largest element `K` in `arr[]` such that `K` is smaller than or equal to `x`.

**Input:** First line of input contains number of testcases `T`. For each testcase, first line of input contains number of elements in the array and element whose floor is to be searched. Last line of input contains array elements.

**Output:** Output the index of floor of `x` if exists, else print `-1`.

### Constraints:

$$1 \leq T \leq 100$$

$$1 \leq N \leq 10^7$$

$$1 \leq \text{arr}[i] \leq 10^{18}$$

$$0 \leq X \leq \text{arr}[n-1]$$

### Input:

```
3
7 0
1 2 8 10 11 12 19
7 5
1 2 8 10 11 12 19
7 10
1 2 8 10 11 12 19
```

### Output:

```
-1
1
3
```

### Explanation:

Testcase 1: No element less than 0 is found. So output is "-1".

Testcase 2: Number less than 5 is 2, whose index is 1(0-based indexing).

### Test Case 1

#### Input

```
1 8 2 22 14 23 10 11 12 19 55
```

#### Output

```
-1
```