Ex.No.:1	PROGRAMS USING BRANCHING AND ITERATIVE CONSTRUCTS

Question 1:

Kumar, the admission incharge in a famous Engineering college wants to write a program to find the eligibility of admission for a professional course based on the following criteria. Marks in Maths >=65, Marks in Phy >=55, Marks in Chem >=50 and Total in all three subject >=180.Help Kumar to write a suitable program.

Input format:

Mathematics, Physics and Chemistry marks in first line separated by space

Output format:

Eligible or Not.

Aim:

To find the eligibility of the student for the admission.

Algorithm:

1.Get the input marks from the user.

2.Using if else statements check the condition for the eligibility.

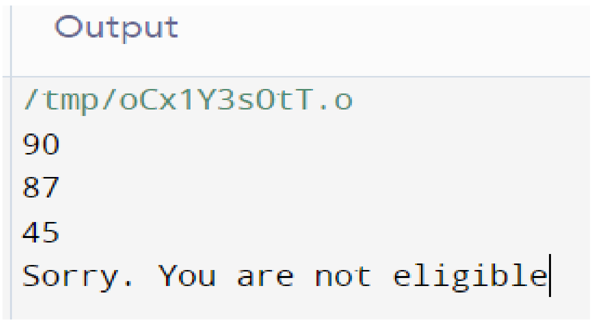
3.Display the output.

Program:

```
#include <iostream>
using namespace std;
int main()
{
   int m,p,c,t;
   cin>>m>>p>>c;
   t=m+p+c;
   if((m>=65) && (p>=55) && (c>=50) && (t>=180))
      cout<<"Congrats. You are eligible";
```

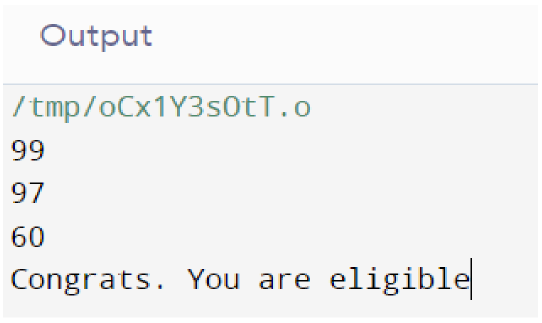
```
        else
            cout<<"Sorry. You are not eligible";
        return 0;
    }
```

Sample Output:



Output

```
/tmp/oCx1Y3sOtT.o
90
87
45
Sorry. You are not eligible
```



Output

```
/tmp/oCx1Y3sOtT.o
99
97
60
Congrats. You are eligible
```

Result:

Thus the program justifies whether the person is eligible or not . The program run successfully and the outputs are verified.

Question 2:

Hanging Bridge

At the annual Kracker Jack Karnival, there was the newest attraction ever in the City, the Hanging Bridge. Visitors will be able to walk 200 ft on the bridge, hanging around 50 ft above the ground, and enjoy a wide-angle view of the breathtaking greenery.

The Hanging Bridge was inaugurated successfully in coordination with the Event Manager Rahul. There is a limit on the maximum number of people on the bridge and Rahul has to now ensure the count of people on the bridge currently should not exceed the limit. He then approximately estimated that C adults and D kids who came to the show, were on the hanging bridge. He also noticed that there are L legs of the people touching the bridge. Rahul knows that kids love to ride on the adults and they might ride on the adults, and their legs won’t touch the ground and hence he would miss counting their legs. Also Rahul knew that the adults would be strong enough to ride at max two kids on their back. Rahul is now wondering whether he counted the legs properly or not. Specifically, he is wondering if there is some possibility of his counting being correct. Please help Rahul in finding it.

Input format

The only line of input contains three space separated integers C, D, L denoting number of the adults, number of the kids and number of legs of people counted by Rahul, respectively.

Output format

Output a single line containing a string “yes” or “no” (both without quotes) according to the situation.

Aim:

To check whether the Ragul’s count and the actual count are equal.

Algorithm:

1.Get the inputs.

2.To find the number of legs, multiply the adults and child count by two.

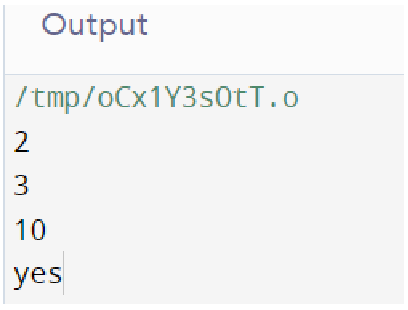
3.Compare the number of legs with the Ragul’s count by using if statement.

4.Display yes if the count matches, else display no.

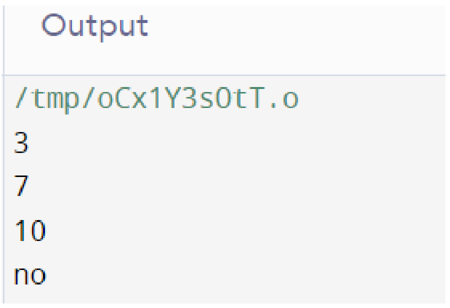
Program:

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    a=(a+b)*2;
    (a==c)?cout<<"yes":cout<<"no";
    return 0;

}
```

Sample Output:



```
Output
/tmp/oCx1Y3sOtT.o
2
3
10
yes
```



```
Output
/tmp/oCx1Y3sOtT.o
3
7
10
no
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 3:

Write a function to get input from the user as hours, minutes, and seconds and convert it into decimal representation in terms of hours, minutes, and seconds. Note: Use suitable function

Input format

Hours: Minutes: Seconds

Output format

Represent the given time in hours, minutes, and seconds

Aim:

To convert the hours:minutes:seconds in hours, minutes and seconds.

Algorithm:

1.Get the input(hour : minute : seconds) in the given format.

2.To find Hour output, convert the minute and seconds input to hour and add it with hour input.

3. To find Minute output, convert the hour and seconds input to minute and add it with minute input.

4. To find Minute output, convert the hour and minute input to seconds and add it with second input.

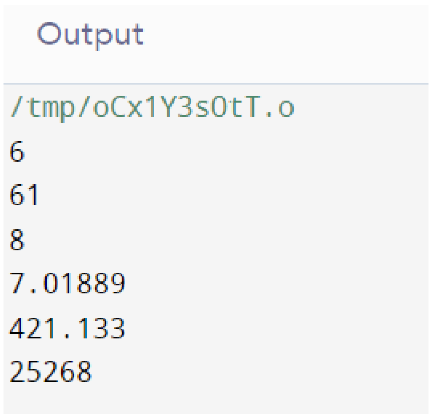
5.Display all the result.

Program:

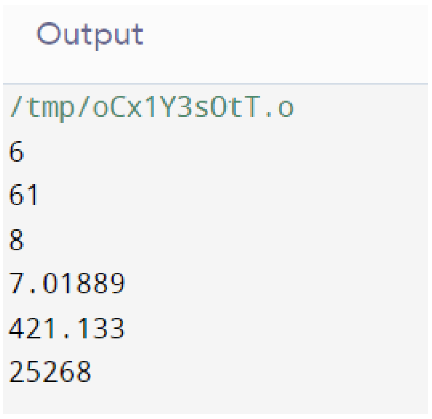
```
#include <iostream>
using namespace std;
int main()
```

```
{
    float hr,min,second;
    cin>>hr>>min>>second;
    float newhr = hr + (min/60.0) + (second/3600.0);
    float newmin = (hr*60) + min + (second/60.0);
    float newsecond = (hr*60*60)+ (min *60)+second;
    cout<<newhr<<endl<<newmin<<endl<<newsecond;
    return 0;
}
```

Sample Output:



```
Output
/tmp/oCx1Y3s0tT.o
6
61
8
7.01889
421.133
25268
```



```
Output
/tmp/oCx1Y3s0tT.o
6
61
8
7.01889
421.133
25268
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 4:

Fibonacci series starts from 0,1 and the next number should be the sum of the previous two numbers.

- 0+1=1, so the next number in the Fibonacci series in the c program is 1.
- 0 1 1 2 3 5 8.. and so on.

Write a program to generate Fibonacci series up to n terms using while loop.

Input format

The input consists of the value of n.

Output format

The output prints the Fibonacci series.

Aim:

To print the Fibonacci series upto the n terms.

Algorithm:

1.Get the input.

2.Set the -1 and 1 as f1 and f2 variable.

3.In a while loop with condition n>0, sum up two variables, and swap f1 with f2 and f2 with f.

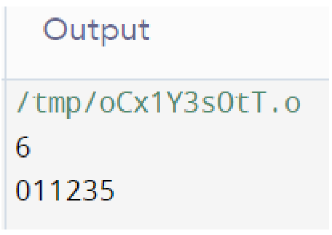
4.Display f and decrement n.

Program:

```
#include <iostream>
using namespace std;
int main()
{
   int n,f,f1=-1,f2=1;
   cin>>n;
```

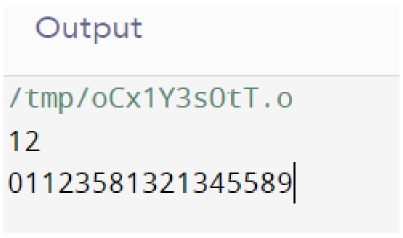
```
    while(n>0)
    {
    f=f1+f2;
    f1=f2;
    f2=f;
    cout<<f;
    n--;
    }
}
```

Sample Output:



Output

```
/tmp/oCx1Y3sOtT.o
6
011235
```



Output

```
/tmp/oCx1Y3sOtT.o
12
01123581321345589
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 5:

Write a program to print the following pattern.

```
7 6 5 4 3 2 1
7 6 5 4 3 2
7 6 5 4 3
7 6 5 4
7 6 5
7 6
7
```

Input format : Number of rows

Output format: Print the pattern as shown in sample output

Aim:

To form inverted right angle triangle with given input.

Algorithm:

1.Get a input .

2. Using two FOR LOOP display the numbers in the descending order and in decreasing way.

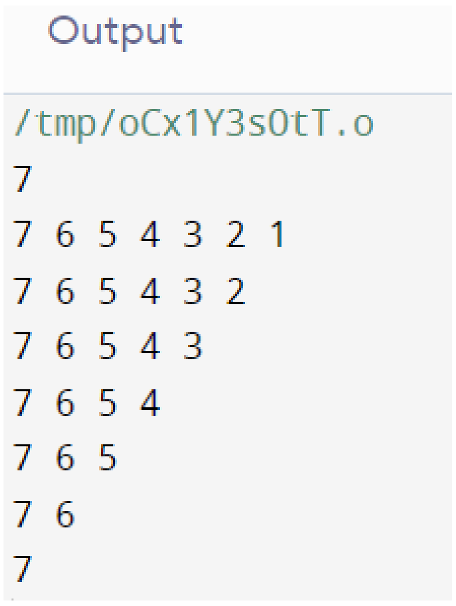
3.Display the output inverted right angle triangle.

Program:

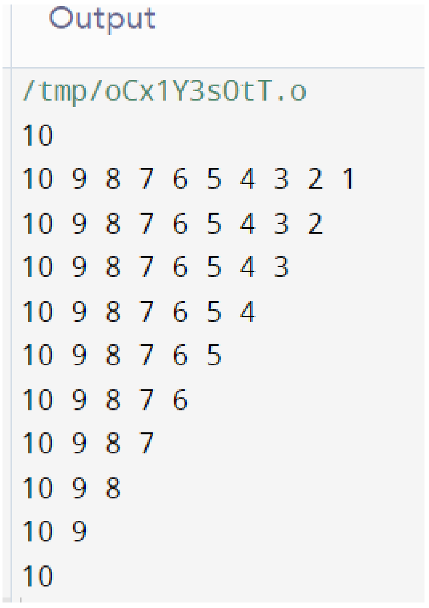
```
#include<iostream>
using namespace std;
int main()
{
   int n,i,j,k;
   cin>>n;
   for(i=1;i<=n;i++)
   {
      for(j=n;j>=i;j--)
      {
```

```
                cout<<j<<" ";
            }
        cout<<endl;
        }
    }
```

Sample Output:



```
Output
/tmp/oCx1Y3sOtT.o
7
7 6 5 4 3 2 1
7 6 5 4 3 2
7 6 5 4 3
7 6 5 4
7 6 5
7 6
7
```



```
Output
/tmp/oCx1Y3sOtT.o
10
10 9 8 7 6 5 4 3 2 1
10 9 8 7 6 5 4 3 2
10 9 8 7 6 5 4 3
10 9 8 7 6 5 4
10 9 8 7 6 5
10 9 8 7 6
10 9 8 7
10 9 8
10 9
10
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.:2	PROGRAMS USING ARRAYS AND STRINGS

Question 1:

Create a dynamic integer array and print the sum over the array of each element * (array element index + 1)

Input format

The first line of input consists of a positive integer n.

The next n lines of input consist of integers that are sequentially assigned to each array element

Output format

The output consists of the sum over the array of each array element * (array element index + 1)

Aim:

Create a dynamic integer array and print the sum over the array of each element.

Algorithm:

1.Get the number of elements in the array and the get the array elements as the input using a for loop.

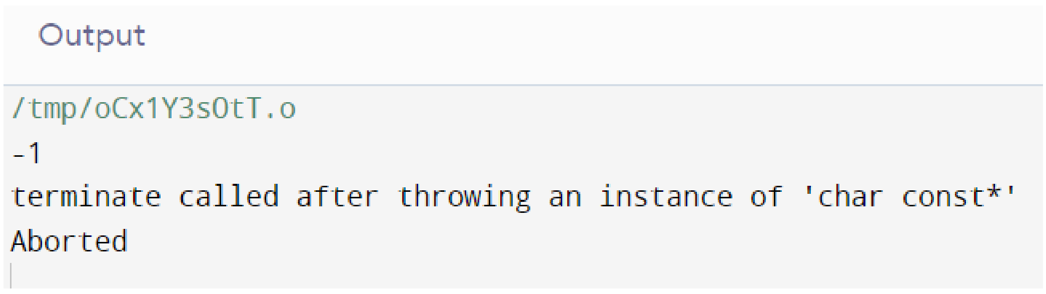
2.Add the multiplied element and its next respective index with a for loop.

3.Display the sum finally.

Program:

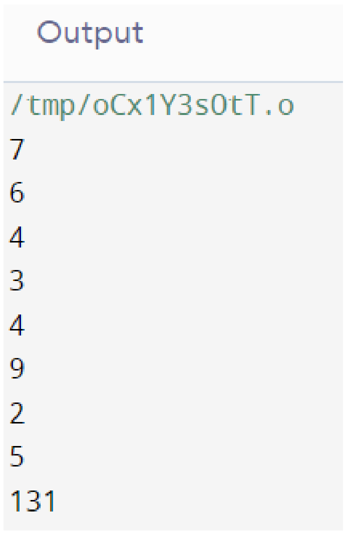
```
#include<iostream>
using namespace std;
int main()
{
        int i, n, sum = 0, count = 0;
        cin >> n;
        if (n < 0)
        {
                throw "input error";
        }
        else
        {
            int *a = new int[n];
            for (i = 0; i < n; i++)
            {
                cin >> a[i];
                sum = sum + (a[i] * (i + 1));
            }
            cout << sum;
        }
    }
```

Sample Output:



Output

```
/tmp/oCx1Y3sOtT.o
-1
terminate called after throwing an instance of 'char const*'
Aborted
```



Output

```
/tmp/oCx1Y3sOtT.o
7
6
4
3
4
9
2
5
131
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

Create a dynamic integer array and print the sum over the array of each element

Most occurring element

An array is filled with one of the values 1, 2, or 3. Find the value that occurs the most in the array. If two or more values occur the most number of times print the lower value.

Input format

The first line of input contains an integer N, denoting the number of elements in the array.

The next line of input contains N values, each of which is either 1, 2, or 3.

Output format

The output prints the minimum number of rooms that need to be painted in order to have all the rooms painted with the same color i.e red, blue, or green.

Aim:

To find the most occurring element in the array.

Algorithm:

1.Get the number of elements in the array and the elements using a for loop.

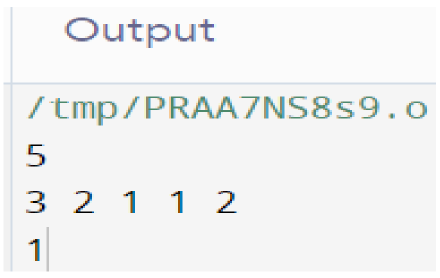
2.By using switch case inside the for loop,the number of 1s,2s,3s are calculated.

3.The maximum occurred element is calculated by the function max and the output is displayed according to it.

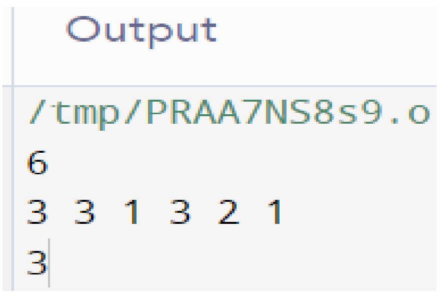
Program:

```
#include<iostream>
using namespace std;
int max(int a, int b){
        if (a > b) return a;
        return b;
}
int main(){
        int n, i, c3, c2, c1;
        cin >> n;
        int *a = new int[n];
        for (i = 0; i < n; i++) {
                cin >> a[i];
        }
        c1 = 0; c2 = 0; c3 = 0;
        for (i = 0; i < n; i++) {
                switch (a[i]) {
                case 1:
                        c1++;break;
                case 2:
                        c2++;break;
                case 3:
                        c3++;break;
                default:
                        throw "invalid input";break;
                }
        }
        int count = max(max(c1, c2), c3);
        if (count == c1) cout << "1";
        else if (count == c2) cout << "2";
        else if (count == c3) cout << "3";
        return 0;
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
5
3 2 1 1 2
1
```



```
Output
/tmp/PRAA7NS8s9.o
6
3 3 1 3 2 1
3
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 3:

Mid Aged : The Pan Am 73 flight from Bombay to New York en route to Karachi and Frankfurt was hijacked by a few Palestinian terrorists at the Karachi International Airport. The senior flight purser Neerja Banhot withered her fear and helped evacuate the passengers on board.

Neerja knew that she would not be able to evacuate all passengers dodging the hijackers. So she wanted to hand over the responsibility of evacuating the senior citizens(above 60 years of age) and children(below 18 years of age) in the flight to the mid-aged passengers seated in the diagonals

Given n, the number of rows of seats and the number of seats in a row and the ages of passengers in each seat can you find the number of mid-aged passengers seated in the diagonals.

Input format

The first line input consists of an integer n, corresponding to the number of rows of seats and the number of seats in the aircraft.

The next n lines of input consist of n integers that correspond to the ages of passengers

Output format

The output consists of an integer corresponding to the number of mid-aged passengers seated in the diagonals.

Aim:

To find the number of mid-aged passengers seated in the seat diagonals.

Algorithm:

1.The number of rows and columns is read as single integer and the inputs of the array is got by two dimensional array using two for loops.

2.Then the number of mid aged persons seated in the diagonal are counted using another two for loops.

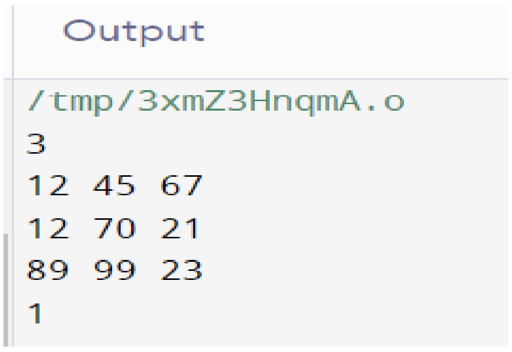
3.The count is displayed finally.

Program:

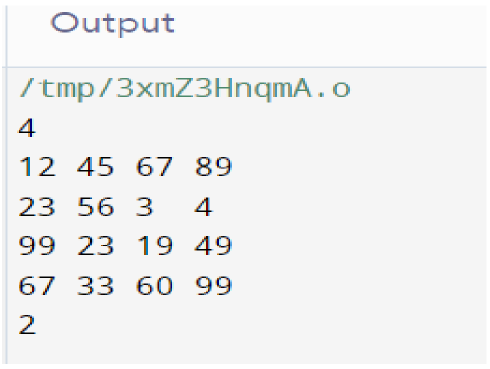
```
#include<iostream>
using namespace std;
int main()
{
   int r,c,i,j,age,count=0;
```

```
                cin>>r;
                int a[r][r];
                for(i=0;i<r;i++)
                {
                        for(j=0;j<r;j++)
                        {
                                cin>>a[i][j];
                        }
                }
                for(i=0;i<r;i++)
                {
                        for(j=0;j<r;j++)
                        {
                           if(i==j)
                           {
                              if(a[i][j]>=18 && a[i][j]<=60)
                              {
                                 count++;
                              }
                           }
                        }
                }
                cout<<count;
        }
```

Sample Output:



```
Output
/tmp/3xmZ3HnqmA.o
3
12 45 67
12 70 21
89 99 23
1
```



```
Output
/tmp/3xmZ3HnqmA.o
4
12 45 67 89
23 56 3  4
99 23 19 49
67 33 60 99
2
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 4:

Number of White Cells

The Nurikabe logical game (sometimes called Islands in the Stream) is a binary determination puzzle. The puzzle is played on a typically rectangular grid of cells, some of which contain numbers. You must decide for each cell if it is white or black (by clicking on them) according to the following rules:

• All of the black cells must be connected.

• Each numbered cell must be part of a white island of connected white cells.

• Each island must have the same number of white cells as the number it contains (including the numbered cell).

• Two islands may not be connected.

• There cannot be any 2x2 blocks of black cells.

Unnumbered cells start out grey and cycle through white and black when clicked. Initially numbered cells are white in color.

Problem Statement:

Write a program to find the number of white cells in the final configuration of the board, given a valid initial configuration. Below figure is the sample valid initial configuration.

Input format

The first line of the input is an integer N that gives the number of rows and columns of the grid.

Next N lines will have a valid initial board configuration with N*N cells. Assume that the maximum number in a cell can be 10.

Grey-colored cells are represented by the integer 20 in the matrix representation of the input configuration.

Output format

The output should display an integer that is the number of white cells in the final configuration of the board.

Aim:

To add the number of white cells.

Algorithm:

1.The number of rows and columns are get as a single integer.

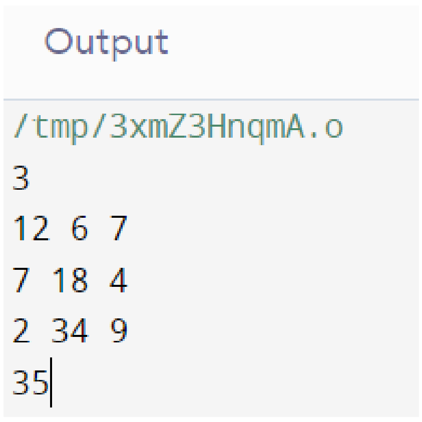
2.Two dimensional array is used to store the elements using two for loops.

3.The sum gives the addition of the matrix elements which is less than 10.

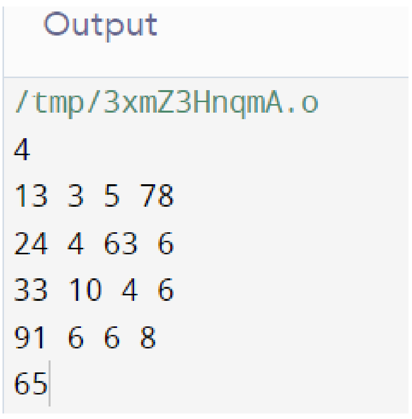
Program:

```
#include<iostream>
using namespace std;
int main()
{
    int n,i,j,k=0;
    cin>>n;
    int a[n][n];
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            cin>>a[i][j];
            if(a[i][j]<11){k=k+a[i][j];}
        }
    }
    cout<<k;
    return 0;
}
```

Sample Output:



```
Output
/tmp/3xmZ3HnqmA.o
3
12 6 7
7 18 4
2 34 9
35
```



```
Output
/tmp/3xmZ3HnqmA.o
4
13 3 5 78
24 4 63 6
33 10 4 6
91 6 6 8
65
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 5:

Adjacent Stick Game

Mukesh and his friends have set out on a vacation to Coorg. They have booked accommodation in a resort and the resort authorities organize Campfires every night as a part of their daily activities. Mukesh volunteered himself for an activity called the "Adjacent Stick Game" where sticks of different lengths will be placed in a line and Mukesh needs to remove a stick from each adjacent pair of sticks. He then has to form a bigger stick by combining all the remaining sticks.

Mukesh needs to know the smallest length of the bigger stick so formed and needs your help to compute the same. Given the number of sticks N and the lengths of each of the sticks, write a program to find the smallest length of the bigger stick that is formed.

Input format

The first line of input contains an integer N denoting the number of sticks. Assume that the maximum value for N is 50.

Assume that N is always even.

The next line of input contains an N integer denoting the length of each of the sticks.

Output format

Output the smallest length of the bigger stick that is formed.

Aim:

To find the smallest length of the bigger stick that is formed.

Algorithm:

1.The number of elements in the array is got as a single integer and the array elements are read using a for loop.

2.A if loop is used to determine the longest sticks of every adjacent sticks.

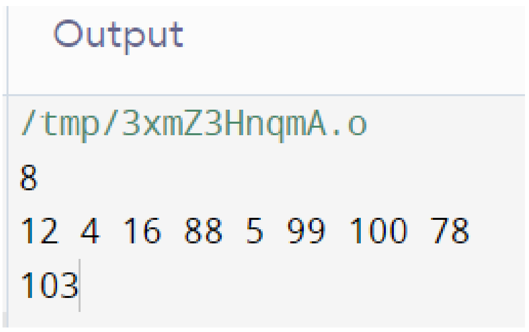
3.Every value of the longer stick is added to a integer and the total addition is displayed finally as the required output.

Program:

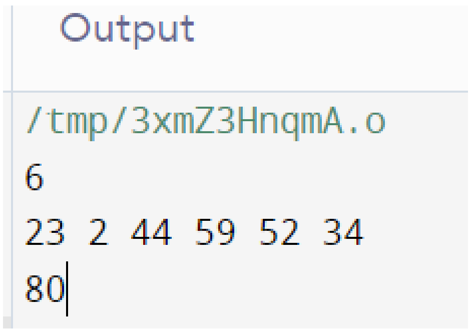
```
#include<iostream>
using namespace std;
int main()
```

```
{
    int n,i,j=0;
    cin>>n;
    int a[n];
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for(i=0;i<n;i=i+2)
    {
        if(a[i]<a[i+1])
        {
            j=j+a[i];
        }
        else
        {
            j=j+a[i+1];
        }
    }
    cout<<j;
    return 0;
}
```

Sample Output:



```
/tmp/3xmZ3HnqmA.o
8
12 4 16 88 5 99 100 78
103
```



```
/tmp/3xmZ3HnqmA.o
6
23 2 44 59 52 34
80
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 6:

Alternating Code

It is IPL season and the first league match of Dhilip's favorite team, "Chennai Super Kings". The CSK team is playing at the IPL after 2 years and like all Dhoni lovers, Dhilip is also eagerly awaiting to see Dhoni back in action.

After waiting in long queues, Dhilip succeeded in getting the tickets for the big match. On the ticket, there is a letter code that can be represented as a string of upper-case Latin letters.

Dhilip believes that the CSK Team will win the match in case exactly two different letters in the code alternate. Otherwise, he believes that the team might lose. Please see the note section for the formal definition of alternating code.

You are given a ticket code. Please determine, whether CSK Team will win the match or not based on Dhilip'sconviction. Print "YES" or "NO" (without quotes) corresponding to the situation.

Note:

Two letters x, y where x != y are said to be alternating in a code if the code is of the form "xyxyxy...".

Input format

The first and only line of the input contains a string S denoting the letter code on the ticket.

Output format

Output a single line containing "Yes" (without quotes) based on the conditions given and "No" otherwise.

Aim:

To print yes if the code is alternating , else to print no.

Algorithm:

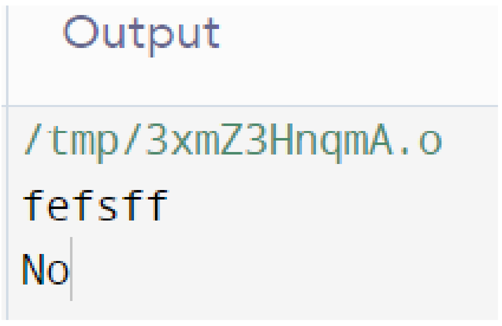
1. The input is got as a array.
2. The array elements are compared with one with the alternating elements using if else condition using for loop.
3. if the sequence fails to satisfy the condition the flaf is raised as 1.
4. The output prints yes if the flag is 0,else it prints No.

Program:

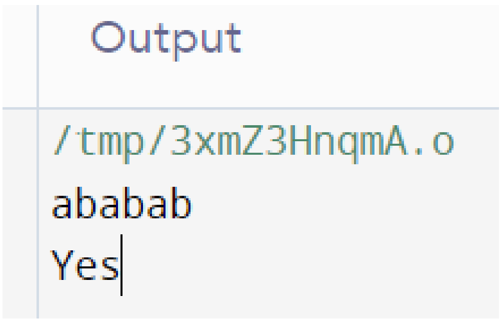
```
#include<iostream>
```

```
#include<cstring>
using namespace std;
int main()
{
    int i,len,s=0;
    char a[20],x,y;
    cin>>a;
    len=strlen(a);
    x=a[0];y=a[1];
    if(x==y){s=1;}
    else{
    for(i=0;i<len;i++)
    {
        if(i%2==0){if(x!=a[i]){s=1;break;}}
        else if(y!=a[i]){s=1;break;}
    }
    }
    (s==1)?cout<<"No":cout<<"Yes";
    return 0;
}
```

Sample Output:



```
Output
/tmp/3xmZ3HnqmA.o
fefsff
No
```



```
Output
/tmp/3xmZ3HnqmA.o
ababab
Yes
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 7:

Wildcard Matching

Sunil is a little scientist. Sunil has planned to design a wildcard pattern matcher to exhibit at the "Young Inventors", a tech talent show organized at his school. Sunil wanted to design the wildcard pattern matcher supporting the wildcard character '?'. The wildcard character '?' can be substituted by any single lower case English letter for matching. He has two strings X and Y of equal length, made up of lower case letters and the character '?'.

Sunil wants your help in designing the device, to know whether the strings X and Y can be matched or not. Write a program to check whether the given strings can be matched or not.

Note : Print 'No' if the length of strings are not equal.

Input format

The first line of the input contains the string ‘X’.

The second line of the input contains the string ‘Y’.

Output format

Output a single line with the word "Yes"(without quotes) if the strings can be matched, otherwise output "No"(without quotes).

Aim:

To design a wildcard pattern matcher with respect to the given conditions.

Algorithm:

1.Two strings are as char arrays.

2.The elements in the index of every array is compared using for loop.

3.If the elements at the respective index are equal or anyone of the elements at that index is “?” , the if loop becomes true ,flag is raised to 1.

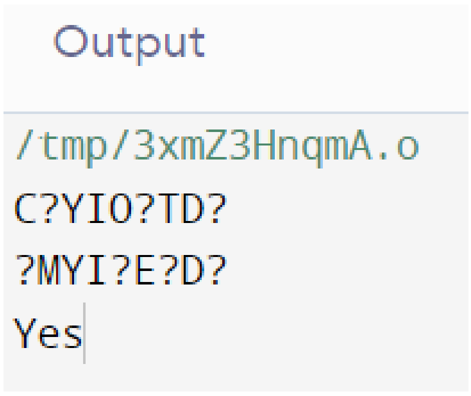
4.At last , according to flag yes or no is displayed by a if and else condition.

Program:

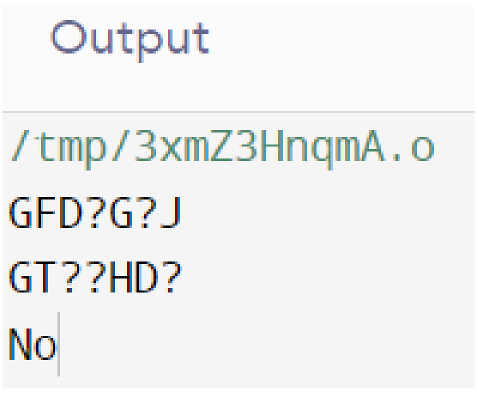
```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    int i,lena,lenb,s=0;
```

```
        char a[50],b[50];
        cin>>a;
        cin>>b;
        lena=strlen(a);
        lenb=strlen(b);
        if(lena==lenb){
        for(i=0;i<lena;i++)
        {
            if(a[i]!=b[i]){if((a[i]!=63)&&(b[i]!=63)){s=1;break;}}
        }
        (s==0)?cout<<"Yes":cout<<"No";
        }
        else
        cout<<"No";
        return 0;
    }
```

Sample Output:



```
/tmp/3xmZ3HnqmA.o
C?YIO?TD?
?MYI?E?D?
Yes
```



```
/tmp/3xmZ3HnqmA.o
GFD?G?J
GT??HD?
No
```

Result:

Thus the program is run successfully and the outputs are verified successfully**.**

Ex.No.:3	PROGRAMS USING FUNCTIONS

Question 1:

Right Triangle of Dots

The much-awaited event in the entertainment industry every year is the "Screen Awards". This year the event is going to be organized on December 25 to honor the Artists for their professional excellence in Cinema. The Organizers of the event, J&R Events, decided to design the logo of the Screen Awards as a digitized image and display it on the LED panel boards for the show promotions all across the venue. The Event team wanted to border the logo with right triangles which will describe it better.
For this purpose, the Event development team is in the task to find if N dots can make a right triangle or not (all N dots must be used). Given N dots, we can make it look like a Right Triangle (45-45-90 triangle) exactly with N dots. Rearrange the given N dots, like this:
Your task is to help the team write a program using functions to find if N dots can make a right triangle or not.

Function Specifications:
Use the function name, return type, and the argument type as:
int find(int)
The function must return 1 if you can make a right triangle using N dots, else return 0.
Note:
The main function is already provided and well defined. The function mentioned above is to be defined by you to solve this problem

Input format
The first line of the input consists of an integer N.

Output format

Output "Yes" (without quotes) if you can make a right triangle using N dots, otherwise "No"(without quotes).

Aim:

To find whether we can create right triangle using given number of dots.

Algorithm:

1.The number of dots is read as a single integer.

2.A function Find() is used to calculate whether it is possible to build a right triangle or not using if conditions inside a for loop.

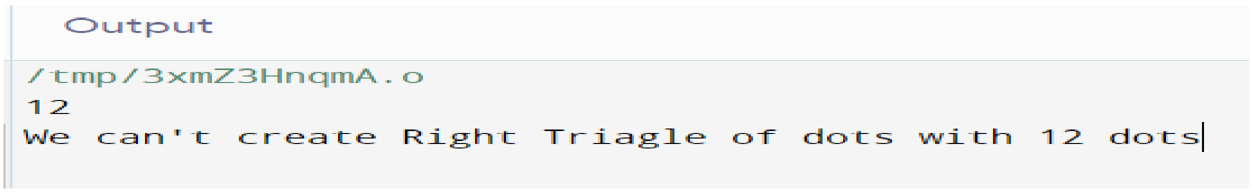
3.The number is got in the main and passed to the function find which displays the respective result.

Program:

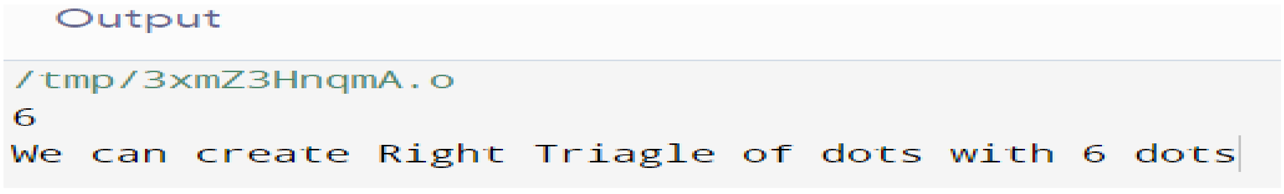
```
#include<iostream>
using namespace std;
int find(int n);
int find(int n)
{
   int i,j=1;
   for(i=1;i<n;i++)
   {
      if((i*(i+1)/2)==n){break;}
      if((i*(i+1)/2)>n){j=0;break;}
   }
   return j;
}
int main()
{
   int n;
```

```
    cin>>n;
    if(find(n)==1){
        cout<<"We can create Right Triagle of dots with "<<n<<" dots";
    }
    else{
        cout<<"We can't create Right Triagle of dots with "<<n<<" dots";
    }
    return 0;
}
```

Sample Output:



```
/tmp/3xmZ3HnqmA.o
12
We can't create Right Triagle of dots with 12 dots
```



```
/tmp/3xmZ3HnqmA.o
6
We can create Right Triagle of dots with 6 dots
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.:4	PROGRAMS USING STRUCTURES AND POINTERS

Question 1:

Create a structure named DEPT with the following fields: Name, emp-id, years_of_experience, and Basic salary. Define an array of structures for 'n' employees and check the following constraints and print the results.

- Increase 10% to the salaries of those employees who have worked for 10 years or more
- Increase 5% to the salaries of those employees who have experienced between 5 to 9 years.
- Increase 2% to the salaries of those employees who have experienced between 1 to 4 years.

Input Format

The first line of the input consists of the value of n.

The next n inputs are the name, id, years of experience, and salary separated by a space.

Output Format

The output prints the employee details with updated salaries.

Aim:

To find the incremented salary of the employee with respect to their experience.

Algorithm:

1.The first input gives the number of employee information.

2.Struct dept is used to get the array of values of the employee.

3.if conditions are used in the For loop in the main function to calculate the salary according to their years of experience.

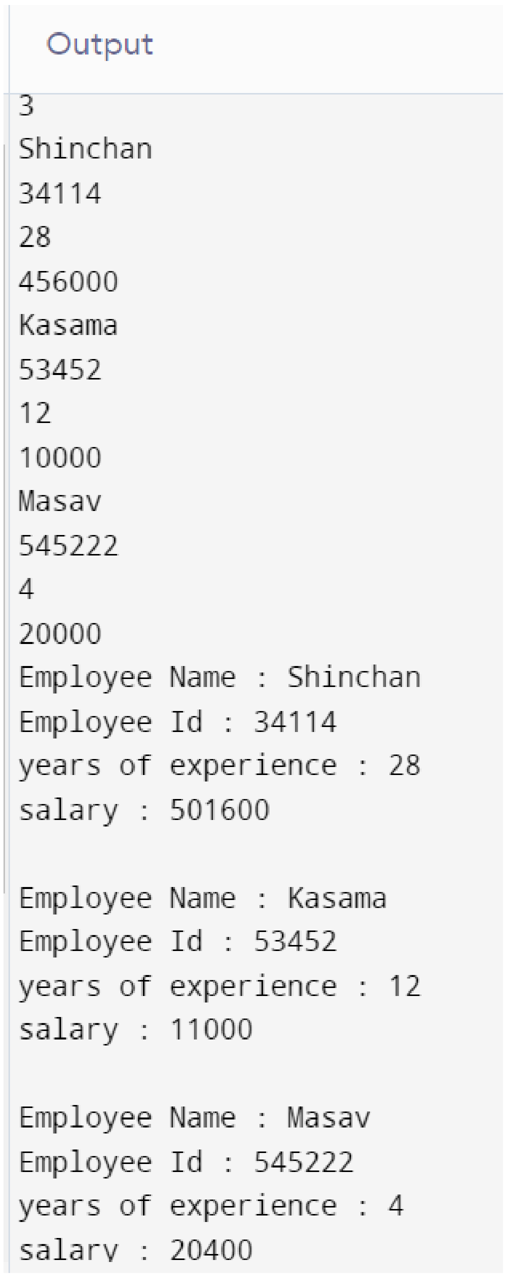
Program:

```
#include<iostream>
```

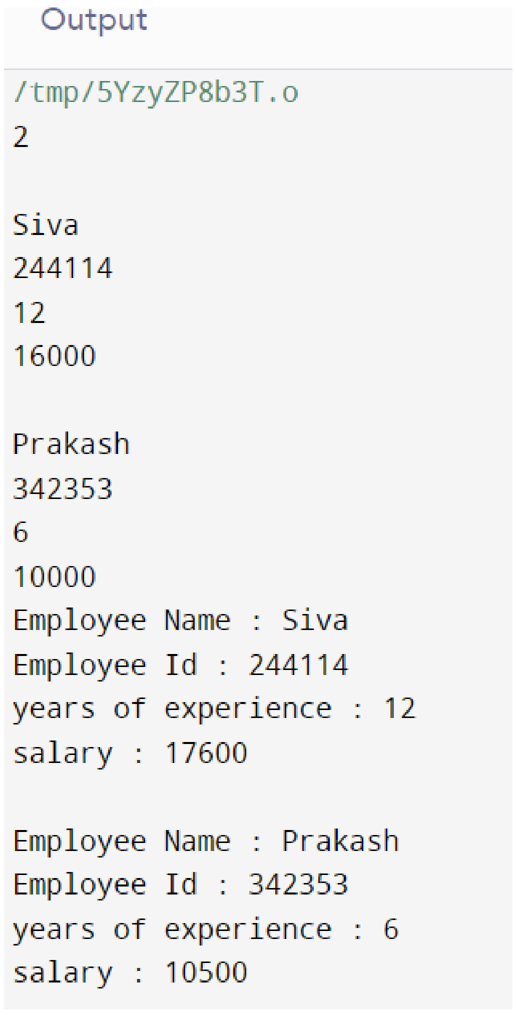
```
using namespace std;
struct DEPT
{
    char name[20];
    int emp_id;
    int yofex;
    int salary;
};
int main()
{
    int n,i;
    cin>>n;
    struct DEPT d[n];
    for(i=0;i<n;i++)
    {
        cin>>d[i].name>>d[i].emp_id>>d[i].yofex>>d[i].salary;
    }

    for(i=0;i<n;i++)
    {
        if(d[i].yofex>=10)
            d[i].salary=(d[i].salary*10)/100+ d[i].salary;
        else if(d[i].yofex>=5 &&d[i].yofex<=9)
           d[i].salary=(d[i].salary*5)/100+ d[i].salary;
        else
            d[i].salary=(d[i].salary*2)/100+ d[i].salary;
    }

    for(i=0;i<n;i++)
    {
        cout<<"Employee Name : "<<d[i].name;
        cout<<"\nEmployee Id : "<<d[i].emp_id;
        cout<<"\nyears of experience : "<<d[i].yofex;
        cout<<"\nsalary : "<<d[i].salary<<endl;
```

```
                cout<<endl;
            }
        }
```

Sample Output:



```
Output
3
Shinchan
34114
28
456000
Kasama
53452
12
10000
Masav
545222
4
20000
Employee Name : Shinchan
Employee Id : 34114
years of experience : 28
salary : 501600

Employee Name : Kasama
Employee Id : 53452
years of experience : 12
salary : 11000

Employee Name : Masav
Employee Id : 545222
years of experience : 4
salary : 20400
```



```
Output
/tmp/5YzyZP8b3T.o
2

Siva
244114
12
16000

Prakash
342353
6
10000
Employee Name : Siva
Employee Id : 244114
years of experience : 12
salary : 17600

Employee Name : Prakash
Employee Id : 342353
years of experience : 6
salary : 10500
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

Create a structure student with the following members

Roll Number

Five subject marks

Average

Grade

Given the five subject marks, Calculate the average and grade.

GRADE CALCULATION:

1)if avg>70 the grade will be 1

2)if avg 50 to 70 the grade will be 2

3)if avg is below 50 the grade will be 3 (**Note:** rn- Roll Number, s-Subjects, avg)

Aim:

To print the average and grade for the given students.

Algorithm:

1.first input gives the number of students.

2.Struct student is used to get the information of array of students.

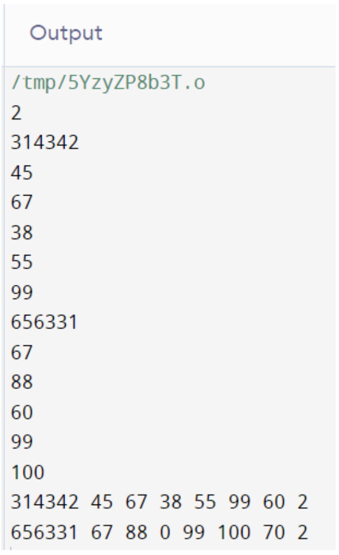
3.If condition is ued inside for loop in main function to display the grade.

Program:

```
#include<iostream>
using namespace std;
struct student
{
int rn,s1,s2,s3,s4,s5,avg,g;
};
int main()
{
  int s,i;
  cin>>s;
  struct student e[s];
  for(i=0; i<s; i++)
  {
  cin>>e[i].rn;
```

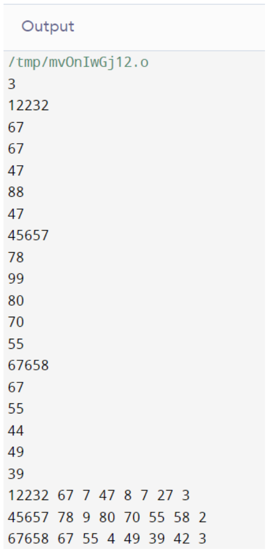
```
    cin>>e[i].s1;
    cin>>e[i].s2;
    cin>>e[i].s3;
    cin>>e[i].s4;
    cin>>e[i].s5;
      e[i].avg = (e[i].s1+e[i].s2+e[i].s3+e[i].s4+e[i].s5)/5;
      if(e[i].avg>70)
        e[i].g = 1;
      else if(e[i].avg>=50 && e[i].avg<=70)
        e[i].g = 2;
      else
        e[i].g = 3;
    }
    for(i=0; i<s; i++){
      cout<<e[i].rn<<"   "<<e[i].s1<<"   "<<e[i].s2<<"   "<<e[i].s3<<"   "<<e[i].s4<<"
  "<<e[i].s5<<" "<<e[i].avg<<" "<<e[i].g<<endl;
    }
    return 0;
  }
```

Sample Output:



Output

```
/tmp/5YzyZP8b3T.o
2
314342
45
67
38
55
99
656331
67
88
60
99
100
314342  45  67  38  55  99  60  2
656331  67  88  0  99  100  70  2
```



Output

```
/tmp/mvOnIwGj12.o
3
12232
67
67
47
88
47
45657
78
99
80
70
55
67658
67
55
44
49
39
12232  67  7  47  8  7  27  3
45657  78  9  80  70  55  58  2
67658  67  55  4  49  39  42  3
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 3:

In an online grocery shop, customers want to purchase multiple items. Create a structure to store the Item code, Brand name, Item Name, Quantity, Price of the product. Generate the Bill number, Display the purchased product, name, amount and quantity, and the total bill amount.

a. Write a function MESSAGE() to alert the customer with the product name if the rate of a product is more than Rs.1000.

b. Write a function VOUCHER() to generate the voucher for Rs.200 if the bill amount is greater than Rs.10000.

Input Format

The first line of the input consists of the value of n.

Next n inputs consist of the item code, brand name, item name, quantity, and price of the product(per item).

Output Format

The output prints a message if the final amount of the product is greater than 1000.

The next line prints the bill amount (Rounded off to two decimal places).

The last line prints whether the customer gets a voucher or not.

Aim:

To display bill along with voucher details.

Algorithm:

1.The first input gives the number of items purchased.

2.Struct grocery is used to get the array of details of the product.

3.Message and voucher details are displayed as the final output using two functions voucher and message.

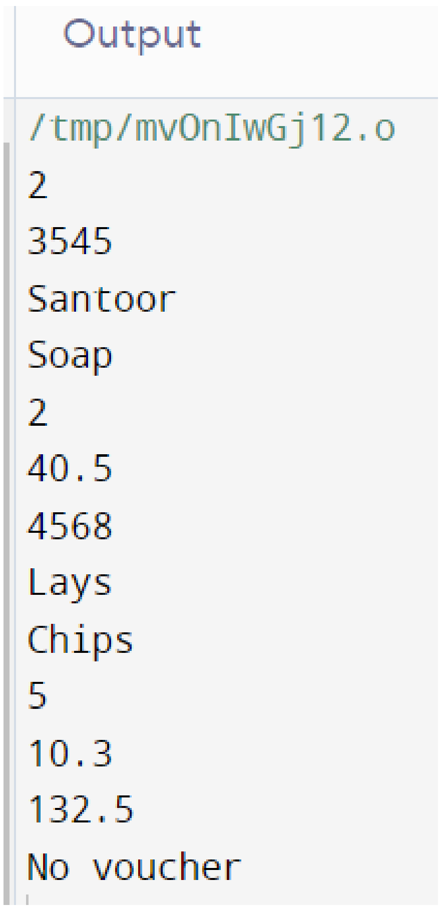
Program:

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

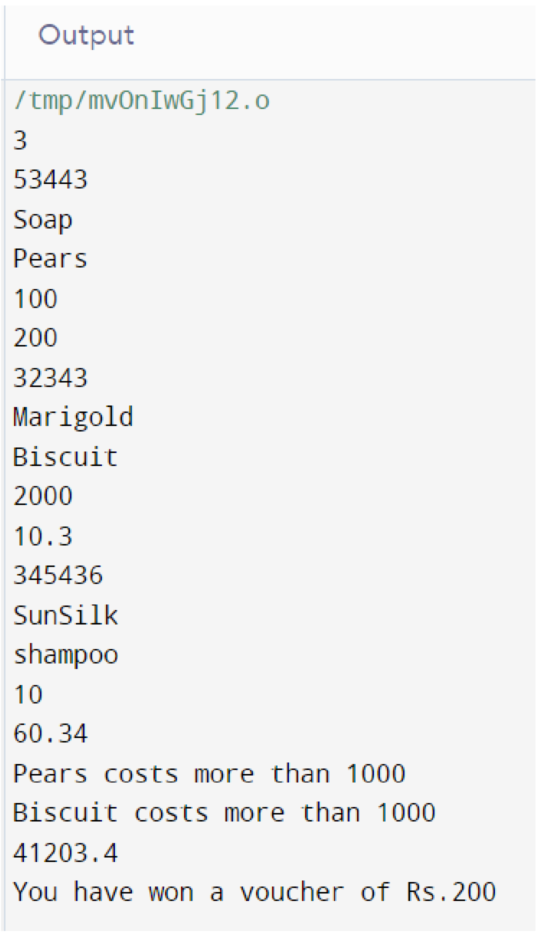
```
{
    int item_code;
    char brand_name[25];
    char item_name[25];
    int quantity;
    float price;
};
void message (struct grocery g[], int n) {
    int i;
    for(i=0;i<n;i++) {
        if(g[i].quantity*g[i].price>1000) {
            cout<<g[i].item_name<<" costs more than 1000\n";
        }
    }
}
void voucher(struct grocery g[], int n) {
    int i;
    float bill=0;
    for(i=0;i<n;i++) {
        bill += (g[i].price*g[i].quantity);
    }
    cout<<bill<<"\n";
    if(bill>10000) {
        cout<<"You have won a voucher of Rs.200\n";
    }
    else {
        cout<<"No voucher\n";
    }
}
int main()
{
    int i,n;
    cin>>n;
    struct grocery g[n];
```

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

    cin>>g[i].item_code>>g[i].brand_name>>g[i].item_name>>g[i].quantity>>g[i].p
    rice;
        }
        message(g,n);
        voucher(g,n);
    }
```

Sample Output:



```
Output
/tmp/mvOnIwGj12.o
2
3545
Santoor
Soap
2
40.5
4568
Lays
Chips
5
10.3
132.5
No voucher
```



```
Output
/tmp/mvOnIwGj12.o
3
53443
Soap
Pears
100
200
32343
Marigold
Biscuit
2000
10.3
345436
SunSilk
shampoo
10
60.34
Pears costs more than 1000
Biscuit costs more than 1000
41203.4
You have won a voucher of Rs.200
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 4:

Write a program to sort a set of strings using pointers.

Input format

The first line of the input consists of the value of n.

The next n inputs are the strings to be sorted.

Output format

The output prints the sorted strings.

Aim:

To sort a set of strings using pointers.

Algorithm:

1.First input gives the number of strings to be reordered.

2.The array of strings are got as pointer array.

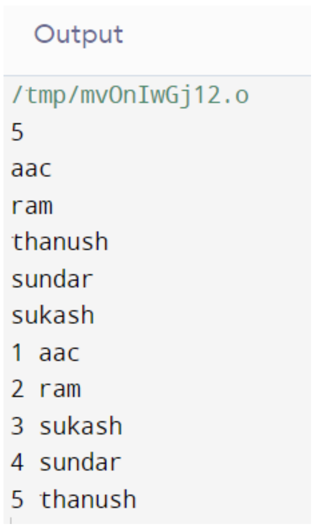
3.The function reorder is used to reorder the strings and the reordered strings are displayed in the output.

Program:

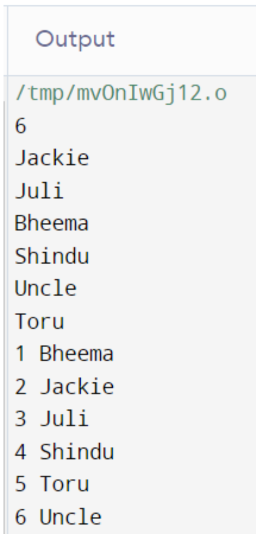
```
#include<iostream>
#include<cstdlib>
#include<string.h>
using namespace std;

int main()
{
    char *x[20];
    int i,n=0;
    void reorder(int n,char *x[]);
    cin>>n;
    for(i=0;i<n;i++)
        {
            x[i]=(char *)malloc(20*sizeof(char));
            cin>>x[i];
        }
            reorder(n,x);
```

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

                              cout<<(i+1)<<" "<<x[i]<<endl;
                         }
        }
        void reorder(int n,char *x[])
        {
           int i,j;
           char t[20];
           for(i=0;i<n-1;i++)
                for(j=i+1;j<n;j++)
                  if(strcmp(x[i],x[j])>0)
                 {
                     strcpy(t,x[j]);
                     strcpy(x[j],x[i]);
                     strcpy(x[i],t);
                 }
                  return;
        }
```

Sample Output:



```
Output
/tmp/mvOnIwGj12.o
5
aac
ram
thanush
sundar
sukash
1 aac
2 ram
3 sukash
4 sundar
5 thanush
```



```
Output
/tmp/mvOnIwGj12.o
6
Jackie
Juli
Bheema
Shindu
Uncle
Toru
1 Bheema
2 Jackie
3 Juli
4 Shindu
5 Toru
6 Uncle
```

Result:

This the program is run successfully and the outputs are verified successfully.

Ex.No.:5	PROGRAMS USING CLASSES AND OBJECTS

Question 1:

Create a class named "dayOfWeek" with d as its private attribute. Use member functions to initialize and print the day of the week.

Aim:

To find the day of the week with respect to its number.

Algorithm:

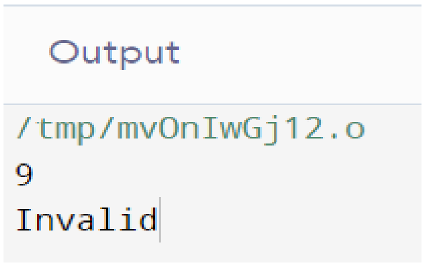
1.The single input is got as input.

2.The input integer is initialized inside the class dayofWeek using the function init and the name of the day is displayed with the use of switch condition inside the function printDay.

Program:

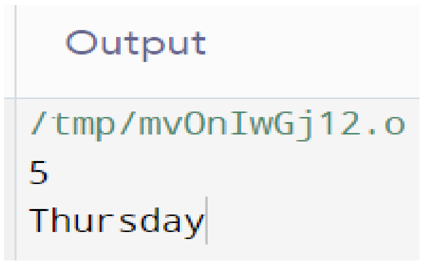
```
#include<bits/stdc++.h>
#include<string>
using namespace std;
class dayOfWeek {
    private:
        int d;
    public:
        void init(int day) {
            d = day;
        }
        void printDay() {
        if(d <= 7){
        switch(d) {
            case 1:
                cout<<"Sunday";break;
            case 2:
                cout<<"Monday"; break;
            case 3:
                cout<<"Tuesday";break;
            case 4:
                cout<<"Wednesday";break;
            case 5:
                cout<<"Thursday";break;
            case 6:
                cout<<"Friday"; break;
            case 7:
                cout<<"Saturday"; break;
           default:
                cout<<"Weekend";break;
        }}
        else{
            cout<<"Invalid";
        }
        }
};
int main() {
    dayOfWeek dow;
    int d;
    cin>>d;
    dow.init(d);
    dow.printDay();
}
```

Sample Output:



Output

```
/tmp/mvOnIwGj12.o
9
Invalid
```



Output

```
/tmp/mvOnIwGj12.o
5
Thursday
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

BMI Calculator

The simple measure of body fitness is the BMI or Body Mass Index. It depends only on the height L and weight W of a person. It is defined as BMI = [weight / height^2] where weight is taken in kilograms and length in meters. Four general grades are proposed:

Underweight[U] - when BMI < 18.5

Normal weight[N] - 18.5 <= BMI < 25.0

Heavyweight [H] - 25.0 <= BMI < 30.0

Overweight [O] - above or equal to 30.

Write a program that takes in the Weight (in Kg) and Length (in meters) of an individual and displays the grade as U, N, H, O.

Note: Bind the data members and the member functions inside the class

Input format

Input to get two values W, L denoting the Weight and the Length. [W is an integer. L is a decimal value].

Output format : print the BMI Grade as U, N, H, or O.

Aim:

To calculate the BMI of a person.

Algorithm:

1.The height and weight are two inputs .

2.BMI is displayed using the function bmi.

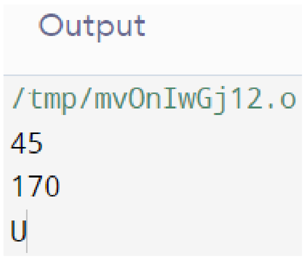
3.bmi function is inside of the class BMI and thus object for BMI is created in main class and the parameters are passed to the function bmi.

Program:

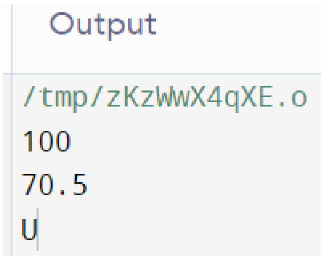
```
#include<iostream>
using namespace std;

class BMI
{
  public:
  void bmi(int n,float m)
  {
  float b;
```

```
    b=n/(m*m);
    if( b <18.5)
      cout<<"U";
    else if (18.5 <= b && b< 25.0 )
      cout<<"N";
    else if(25.0 <= b && b< 30.0)
      cout<<"H";
    else
      cout<<"O";
    }
  };

  int main()
  {
        int n;
        cin>>n;
        float m;
        cin>>m;
        BMI t ;
        t.bmi(n,m);

  }
```

Sample Output:



```
Output
/tmp/mvOnIwGj12.o
45
170
U
```



```
Output
/tmp/zKzWwX4qXE.o
100
70.5
U
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.:6	PROGRAMS USING CONSTRUCTOR AND DESTRUCTOR

Question 1:

Write a program to find the square, square root, cube, and cube root of the given number.
Create a class Demo and include a constructor to initialize the value and write a method display to print square, square root, cube, and cube root of the given number after that the destructor destroys the object.

Aim:

To find the square, square root, cube, and cube root of the given number.

Algorithm:

1.The input is a single integer.

2.The input is passed as parameter to the function display which is inside of the class Demo.

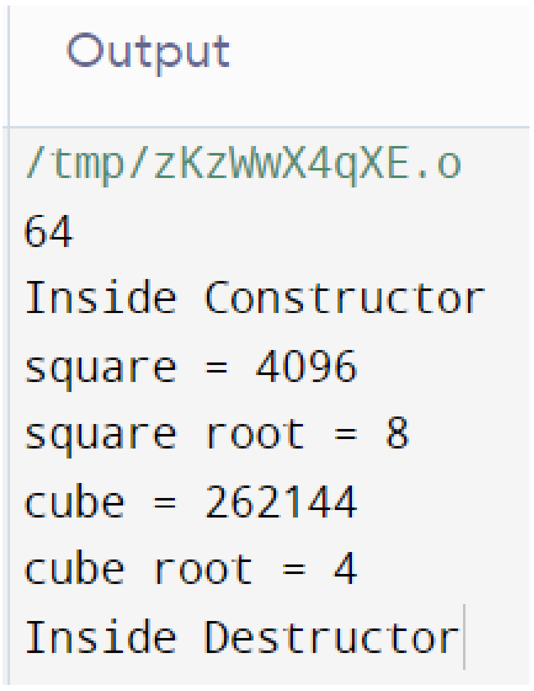
3.Hence, at first , an object is created for the class Demo and thus the output is achieved.

Program:

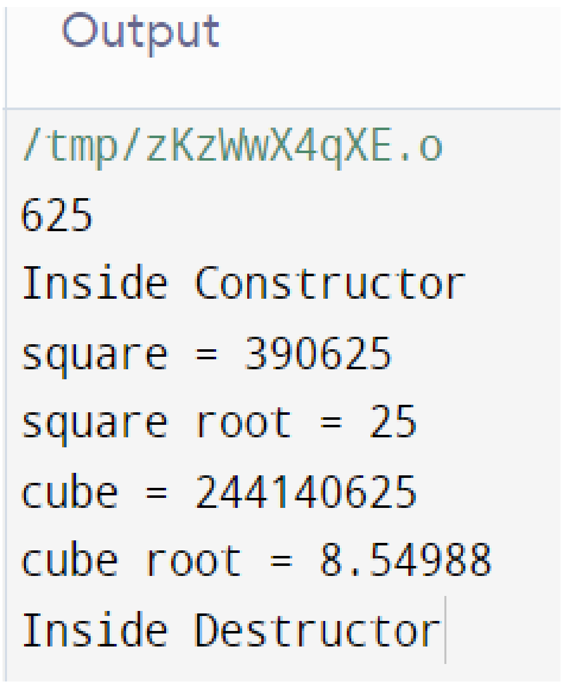
```
#include <bits/stdc++.h>
#include<cmath>
using namespace std;
class Demo {
   private:
   int n;
   public:
   Demo(int num) {
      cout<<"Inside Constructor"<<endl;
      n=num;
   }
   void display() {
      cout<<"square = "<< n*n <<endl;
```

```
            cout<<"square root = "<< sqrt(n) <<endl;
            cout<<"cube = "<<n*n*n<<endl;
            cout<<"cube root = "<<cbrt(n)<<endl;
        }
        ~Demo() {
            cout<<"Inside Destructor";
        }
    };
    int main() {
        int n;
        cin>>n;
        Demo obj1(n);
        obj1.display();
        return 0;
    }
```

Sample Output:



```
Output

/tmp/zKzWwX4qXE.o
64
Inside Constructor
square = 4096
square root = 8
cube = 262144
cube root = 4
Inside Destructor
```



```
Output

/tmp/zKzWwX4qXE.o
625
Inside Constructor
square = 390625
square root = 25
cube = 244140625
cube root = 8.54988
Inside Destructor
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.:7	PROGRAMS USING METHOD OVERLOADING AND OPERATOR OVERLOADING

Question 1:

Write a program to implement function overloading.

Ram is given two or three inputs as an integer, if he has two integers then add the two numbers. If he has three inputs, then multiply the three numbers.

Function Header:

public void fun1(int a ,int b , int c)

public void fun1(int a ,

int b)

Input Format

First-line represents the number of elements(N) followed by the elements separated by a single space. If the number of the elements exceeds 2 or 3, then display a message as Invalid Input.

Output Format

Display the sum, if there are two integers or Displays product, if there are three integers.

Constraints

N > 0 and N < 4

Aim:

To display the sum and product of the numbers with respect to the conditions.

Algorithm:

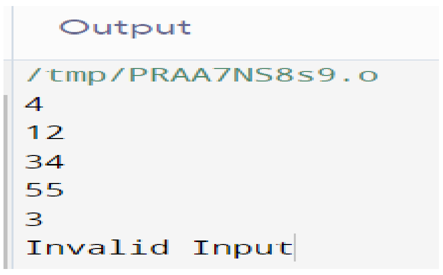
1.First input denotes the number of integers.
2.function fun1 is used inside the class Fun .
3.Object is created for the class Fun in the main class.
4.if first input integer is 2,two parameter are passed to fun1 function or if the input is 3, three parameters are passed to the fun1 function.

Program:

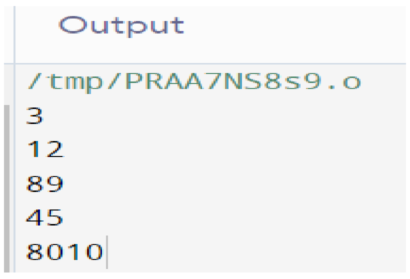
```
#include <bits/stdc++.h>
using namespace std;
```

```
class Fun {
public:
    int fun1(int num1,int num2) {
        return num1+num2;
    }
    int fun1(int num1,int num2, int num3) {
       return num1*num2*num3;
    }
};
int main() {
    Fun obj;
    int i,n;
    cin>>n;
    int arr[n];
    for(i=0;i<n;i++)
        cin>>arr[i];
    if(n==2)
        cout<<obj.fun1(arr[0],arr[1]);
    else if(n==3)
        cout<<obj.fun1(arr[0],arr[1],arr[2]);
    else
        cout<<"Invalid Input";
    return 0;
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
4
12
34
55
3
Invalid Input
```



```
Output
/tmp/PRAA7NS8s9.o
3
12
89
45
8010
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

An ice-cream vendor sells his ice-creams in a cone (radius r and height h) and square(side r) shaped containers. However, he is unsure of the quantity that can be filled in the two containers. You are required to write a program that prints the volume of the containers given their respective dimensions as input. Your class must be named ‘Icecream’ which has two methods with the same name ‘Quantity’ each having the respective dimensions of the containers as the parameters.

Function Header:

void Quantity(int r, int h)

void Quantity(int r)

Formulas:

Volume of a cone = 0.33*pi*r*r*h.

Volume of a square = r*r*r.

Input Format

The first line of the input consists of the choice (1 for square, 2 for cone).

If the choice is 1, Enter the side of the square.

If the choice is 2, Enter the radius and height of the cone separated by a space.

Note: Input type should be an integer.

Output Format

The output must display the volume of the container rounded off to two decimal places for which the dimensions are given.

Aim:

To display the volume of the container rounded off to two decimal places for which the dimensions are given.

Algorithm:

1.The first input integer gives the shape.

2.According to the integer , measurements are taken.

3.Object is created for the class Icecream .

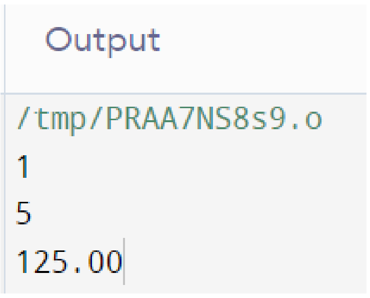
4.The measurements are passed as parameter to the function quantity which is inside of the class Icecream.

Program:

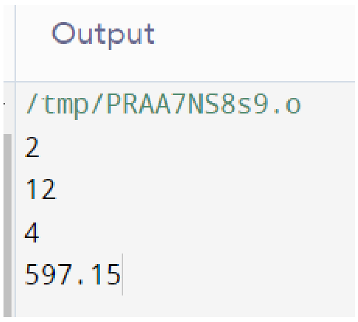
```
#include <bits/stdc++.h>
#include<cmath>
```

```
using namespace std;
class Icecream {
public:
    float Quantity(int r) {
        return r*r*r;
    }
    float Quantity(int r, int h) {
       return 0.33*M_PI*r*r*h;
    }
};
int main() {
    Icecream obj;
    int i,n,r,h;
    cin>>n;
    if(n==1) {
        cin>>r;
        cout<<fixed<<setprecision(2)<<obj.Quantity(r);
    }
    if(n==2) {
        cin>>r>>h;
        cout<<fixed<<setprecision(2)<<obj.Quantity(r,h);
    }
    return 0;
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
1
5
125.00
```



```
Output
/tmp/PRAA7NS8s9.o
2
12
4
597.15
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 3:

C++ provides an option to overload operators. When placed between integer operands, a single operator '+' adds them and concatenates them when placed between string operands. Write a program to overload the operator('+') to compute the sum of complex numbers.

Aim:

To add imaginary numbers.

Algorithm:

1.The 4 inputs are read which are the real and imaginary parts of the two complex numbers.

2.+ operator is used to add the imaginary and real part in the class complex.

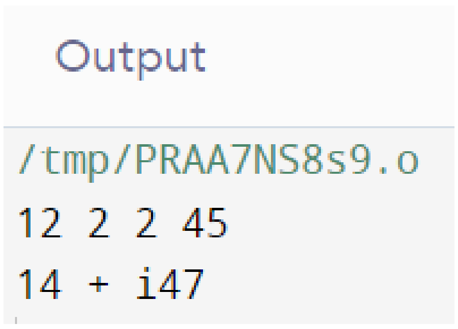
3.Finally the output is displayed by the function print in class complex.

Program:

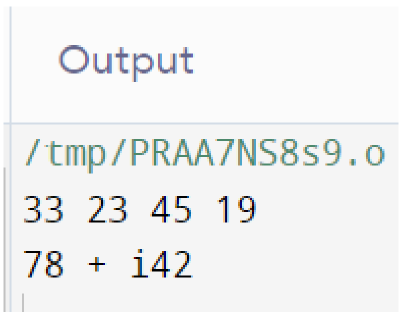
```
#include<iostream>
using namespace std;
class Complex {
private:
        int real, imag;
public:
        Complex(int r = 0, int i =0) {
           real = r; imag = i;
        }
        Complex operator + (Complex const &obj) {
                Complex res;
                res.real = real + obj.real;
                res.imag = imag + obj.imag;
                return res;
        }
        void print() {
           cout << real << " + i" << imag << endl;
```

```
            }
    };

    int main()
    {
       int r1,i1,r2,i2;
       cin>>r1>>i1>>r2>>i2;
            Complex c1(r1, i1), c2(r2, i2);
            Complex c3 = c1 + c2;
            c3.print();
    }
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
12 2 2 45
14 + i47
```



```
Output
/tmp/PRAA7NS8s9.o
33 23 45 19
78 + i42
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.:8	PROGRAMS USING FRIEND FUNCTION

Question 1:

Create a class Rectangle with length and breadth as its attributes along with a constructor. Declare a friend function
void calcArea(sample s) - prints the area of the rectangle.
In the main method, create an object for the rectangle class and call the methods.

Aim:

To find the area of the rectangle.

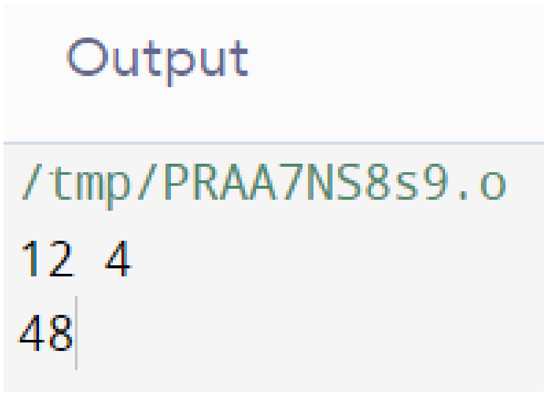
Algorithm:

1.Length and breadth of the rectangle is read through the main function.
2.The parameters are passed to the constructor Rectangle of class Rectangle.
3.The function calcArea is declared as friend function inside the class.
4.The output is displayed through the function by passing the object created in the main class to the function calcArea.

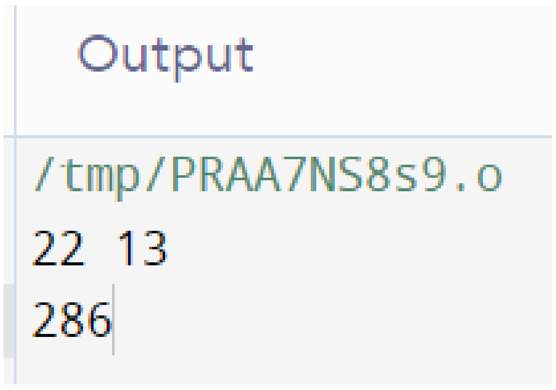
Program:

```
#include <bits/stdc++.h>
#include <string>
using namespace std;
class Rectangle{
  int length, breadth;
  public:
  Rectangle(int length, int breadth):length(length),breadth(breadth)
  {}
  friend void calcArea(Rectangle s);
};
void calcArea(Rectangle s){
    cout<<s.length * s.breadth;
    }
  int main()
    {
      int len,bre;
      cin>>len>>bre;
      Rectangle s(len,bre);
      calcArea(s);
      return 0;
  }
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
12 4
48
```



```
Output
/tmp/PRAA7NS8s9.o
22 13
286
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

Create two classes A and B with x and y as their private attributes. Both classes have a public member function,

setdata(int) - Assign the values to x and y

Declare a friend function in both the classes - void min(int,int) - prints the minimum of x and y. Create a driver class main and access the methods.

Aim:

To print the minimum value among two numbers.

Algorithm:

1.The two inputs are got as integer in the main function and 1st number is passed to class A and 2nd number is passed to class B . function min is set as friend for both the classes.

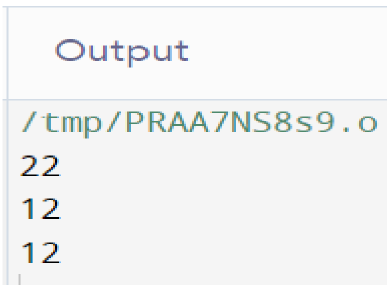
2.By creating objects to the classes, parameters are passed into the function.

3.Minimum of the numbers is displayed finally.

Program:

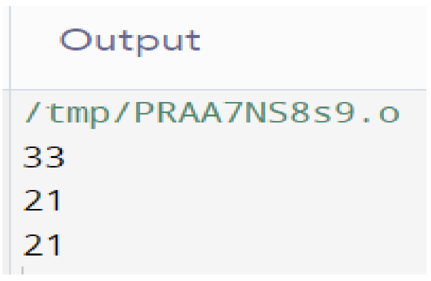
```
#include <bits/stdc++.h>
using namespace std;
class B;
class A  {
   int x;
   public:
   void setdata(int i)
   {
      x=i;
   }
   friend void min(A,B);
};
class B  {
   int y;
   public:
    void setdata(int i)
    {
        y=i;
    }
    friend void min(A,B);
};
void min(A a,B b)  {
    if(a.x<=b.y)
    std::cout << a.x << std::endl;
    else
    std::cout << b.y << std::endl;
}
   int main()  {
   A a;
   B b;
   int x,y;
   cin>>x>>y;
   a.setdata(x);
   b.setdata(y);
   min(a,b);
    return 0;
 }
```

Sample Output:



Output

```
/tmp/PRAA7NS8s9.o
22
12
12
```



Output

```
/tmp/PRAA7NS8s9.o
33
21
21
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 9	PROGRAMS USING VIRTUAL FUNCTIONS & ABSTRACT CLASS

Question 1:

Write a program to find whether the given number(x) is even or odd if it is even then print the cube(x)+square(x) else print cube(x)-square(x). Create a base class with a virtual function void print(). print the result by implementing this virtual function in the derived class.

Input Format

The input consists of an Integer.

Output Format

The output prints the result.

Aim:

To print sum of cube of the number and the square of the number if the number is even , else to print the difference between cube of that number and square of that number.

Algorithm:

1.The input is got a single integer in the main function.

2.Virtual function print is created in class A.

3.In class B , function EvenorOdd finds whether the number is even or odd and do the calculations using a block of if conditions.

4.Function print in the class B is used to print the final output.

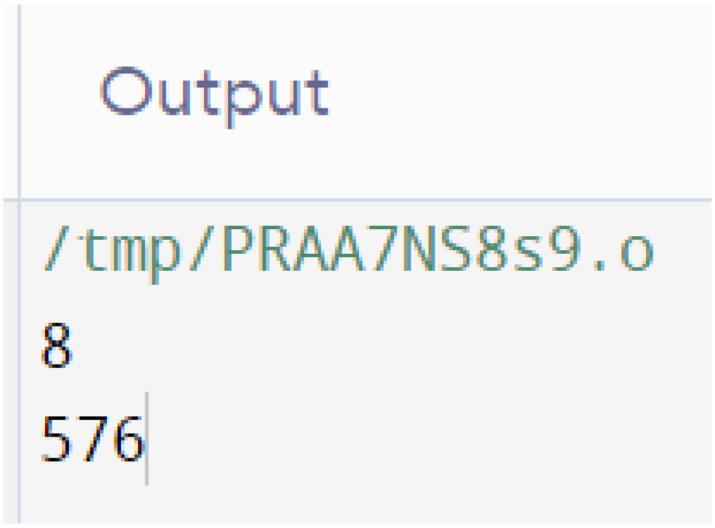
Program:

```
#include <iostream>
using namespace std;
class A {
    public:
```

```
    virtual void print()=0;
};
class B: public A{
    public:
    int x=9;
    int EvenOrOdd(int a) {
        if(a%2==0) return 1;
        else return 0;
    }
    int cube(int n) {
        return (n*n*n);
    }
    int square(int n) {
        return (n*n);
    }
    int sum(int a,int b){
        return (a+b);
    }
    int sub(int a,int b) {
        return (a-b);
    }
    void print() override{
        cout<<x;
    }
};
int main()
{
   int n;
   cin>>n;
   B ob;
   if(ob.EvenOrOdd(n)==1){
       ob.x=ob.sum(ob.cube(n),ob.square(n));
       ob.print();
   }
```

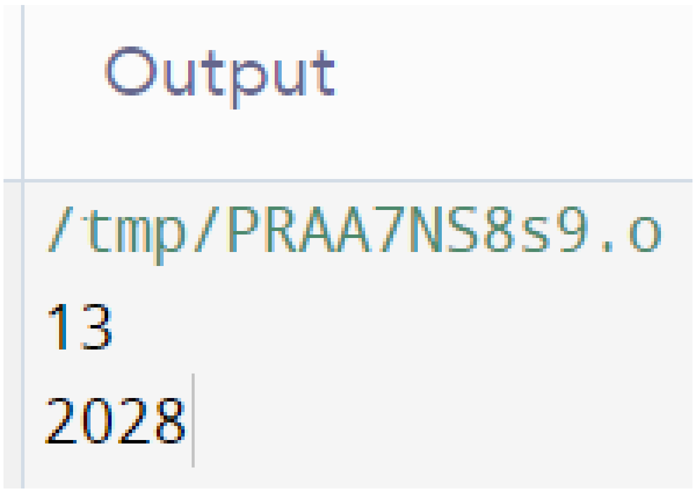
```
        else{
            ob.x=ob.sub(ob.cube(n),ob.square(n));
            ob.print();
        }
        return 0;
    }
```

Sample Output:



Output

```
/tmp/PRAA7NS8s9.o
8
576
```



Output

```
/tmp/PRAA7NS8s9.o
13
2028
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

You have to enter four letters for each uppercase letter you will get 10 marks and for each lowercase letter, you will get -5 marks. Write a program to calculate the total score.

Create a base class with a virtual method void game(). Define this method in the Derived class and calculate the total score.

Input Format

Input consists of four characters separated by space.

Output Format

The output prints the total score.

Aim:

To calculate the total score.

Algorithm:

1.The input consists of four characters.

2.The program consists of two classes namely Base which contains virtual function void game() and Class Derived which is a child of Base.

3.value of 10 is added to upper character and value of -5 is subtracted for lower character .Thus the final output gives the total score.

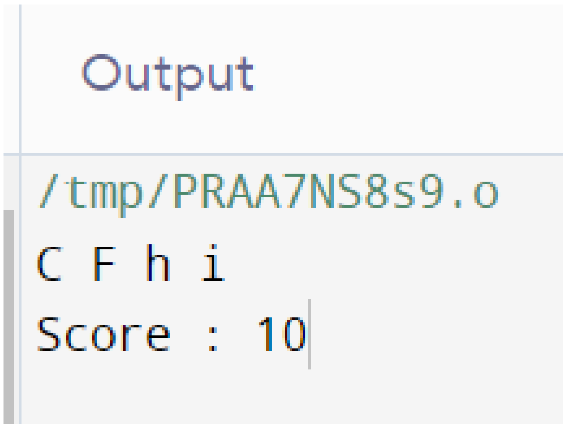
Program:

```
#include <iostream>
using namespace std;
class Base{
   public:
      virtual void game()=0;
};
class Derived: public Base{
   public:
   char a,b,c,d;
```

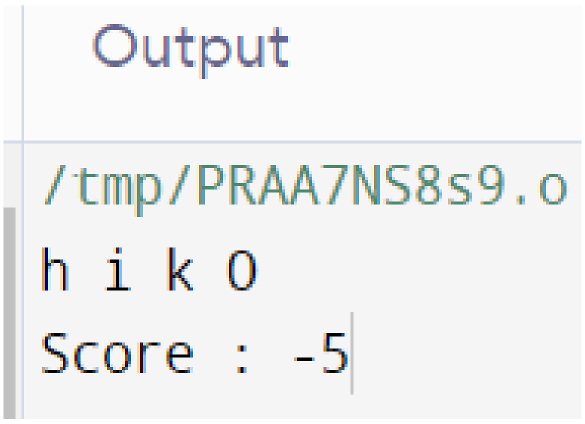
```
    int s=0;
    void game() override{
        if(a>='A'&&a<='Z'){
           s=s+10;
        }
        else{
            s=s-5;
        }
        if(b>='A'&&b<='Z'){
           s=s+10;
        }
        else{
            s=s-5;
        }
        if(c>='A'&&c<='Z'){
           s=s+10;
        }
        else{
            s=s-5;
        }
        if(d>='A'&&d<='Z'){
           s=s+10;
        }
        else{
            s=s-5;
        }
        cout<<"Score : "<<s;
    }
};

int main()
{
  Derived dd;
   cin>>dd.a>>dd.b>>dd.c>>dd.d;
```

```
        dd.game();

        return 0;
    }
```

Sample Output:



```
/tmp/PRAA7NS8s9.o
C F h i
Score : 10
```



```
/tmp/PRAA7NS8s9.o
h i k O
Score : -5
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 10	PROGRAMS USING INHERITANCE CONCEPTS

Question 1:

Write a program to implement the following logic using inheritance. Create a parent class and implement the fun method. In the method, get the individual digits of the entered number, store it in an array, and find their sum. Create the main class that inherits the parent class and call the fun method inside the parent function.

Example

Input

1234

Output

30

Explanation

For 1234, the individual digits are 4,3,2,1 and the final sum → (4+3)+(4+2)+(4+1)+(3+2)+(3+1)+(2+1) = 30.

Input Format: The input consists of an integer.

Output Format: The output prints the final sum.

Constraints: Integers only.

Aim:

To, get the individual digits of the entered number, store it in an array, and find their sum.

Algorithm:

1.The input is got as a single integer using the function read() which belongs to the class child which is the child of the class Parent.

2.Then the function fun() is called in the main function , whereas in fun() , fun1() function of the parent class is called.

3.For loop of the fun() function calculates the final sum of the given integer.

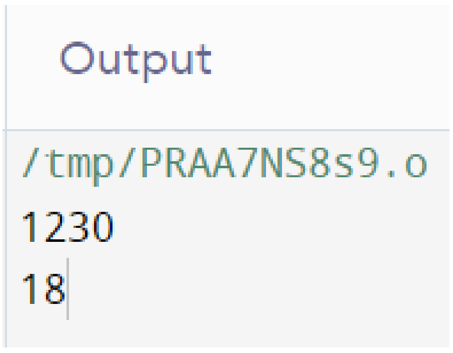
4.Objects of derived class is created in main function and each functions are called.

Program:

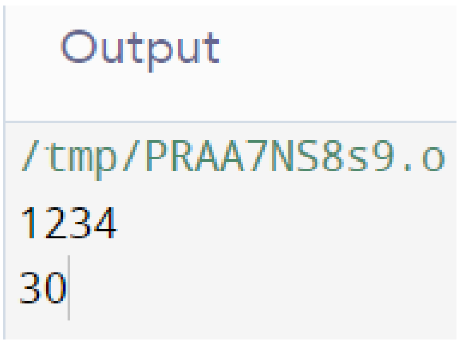
```
#include<iostream>
using namespace std;
class parent
{
   public:
   int num;
   void fun()
   {
      int t,i,j,a[100],m,k=0;
      t=num;
      while (t!=0)
      {
         m=t%10;
         a[k++]=m;
         t=t/10;
      }
      if(k==1){
         cout<<a[0];
         return;
      }
      int sum=0;
      for(i=0;i<k;i++)
      {
         for(j=i+1;j<k;j++)
         {
         sum=sum+a[i]+a[j];
         }
      }
      cout<<sum;
   }
};
```

```
class child:public parent
{
    public:
    void read()
    {
        cin>>num;
    }
    void fun1()
    {
        fun();
    }
};
int main()
{
    child c;
    c.read();
    c.fun1();
 }
```

Sample Output:



```
/tmp/PRAA7NS8s9.o
1230
18
```



```
/tmp/PRAA7NS8s9.o
1234
30
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 11	PROGRAMS USING EXCEPTION HANDLING CONCEPT

Question 1:

Check whether the person is eligible for a driving license or not. If the age is less than 18, throw an exception and print "Invalid Age" and quit. If the score is less than 40, throw an exception and print "You should get at least 40 marks" and quit. Else print "Passed".

Input format

The input consists of the age and score separated by a space.

Output format

The output prints the age, score, and result.

Aim:

To check whether the person is eligible for a driving license or not.

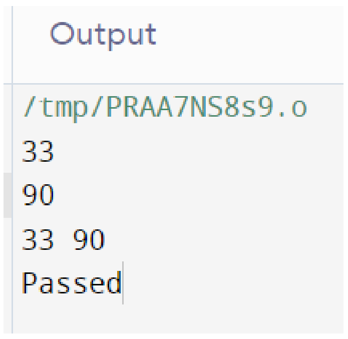
Algorithm:

1. Two inputs are read in the main function.
2. If the age is less than 18, the age is thrown into catch which catches int and displays the content of it.
3. If the score is less than 18, the character is thrown into catch which catches char and displays the content of it.
4.

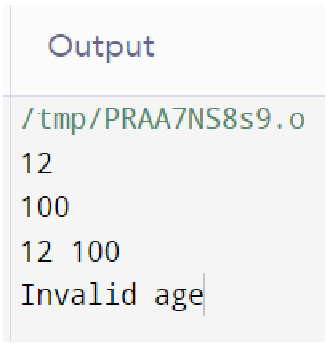
Program:

```
#include<iostream>
using namespace std;
int main()
{
int a,b;
cin>>a>>b;
try
{
  if(a>18&&b>=40)
  cout<<a<<" "<<b<<endl<<"Passed";
  else if(a<18)
  throw a;
  else if(b<40&&a>18)
  throw 'b';
}
catch (int s)
{
    cout<<a<<" "<<b<<endl<<"Invalid age";
}
catch (char ch)
{
    cout<<a<<" "<<b<<endl<<"You should get atleast 40 marks";
}
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
33
90
33 90
Passed
```



```
Output
/tmp/PRAA7NS8s9.o
12
100
12 100
Invalid age
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

Create a vector of n size and add elements to it. Get a position and a value and modify the same. If the position exceeds the size of the vector, throw and catch an exception and print the same.

Input Format

The first line of the input consists of the n value.

The next line of the input consists of the values of the vector.

The last line of the input consists of the position and the element to be replaced.

Output Format

The output prints the modified/original vector and the exception.

Aim:

To replace the given integer at the given index in the given vector and find its sum.

Algorithm:

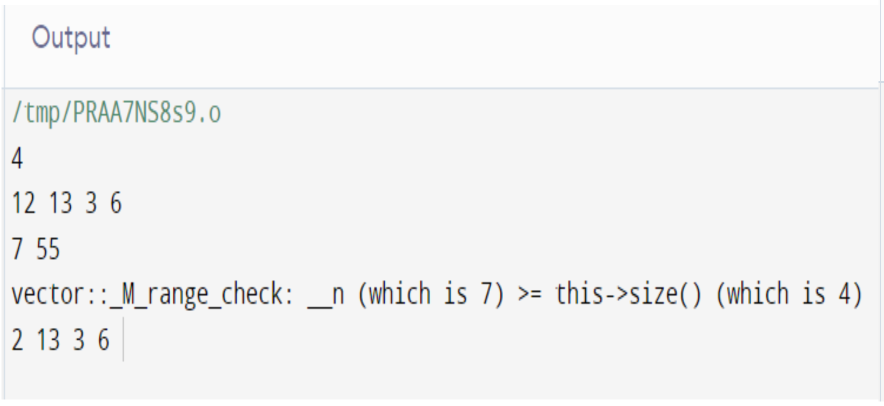
1. First input is the number of array elements and the following array of elements are read using a for loop.
2. Next line of input is the index and the number to be replaced.
3. If the given index is greater than that of the size of the array, it throws the index, which is caught by catch of integer type and displays its content.
4. If index is less than the size of array , for loop is used to check the index value and when condition is satisfied, it replaces the older element with new element.

Program:

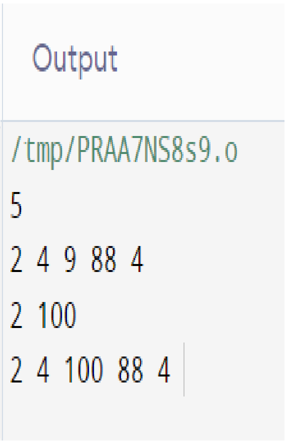
```
#include<iostream>
using namespace std;
int main() {
    int n,pos,val;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++) {
        cin>>a[i];
    }
    cin>>pos>>val;
    try {
        if(pos>n)
        throw 1;
```

```
            else
            for(int i=0;i<pos+1;i++)  {
                a[pos]=val;
            }
            for(int i=0;i<n;i++)   {
                cout<<a[i]<<" ";
            }
        }
        catch (int p)    {
            cout<<"vector::_M_range_check: __n (which is "<<pos<<") >= this->size()
    (which is "<<n<<")"<<endl;
            for(int i=0;i<n;i++)    {
                cout<<a[i]<<" ";
            }

        }
    }
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
4
12 13 3 6
7 55
vector::_M_range_check: __n (which is 7) >= this->size() (which is 4)
2 13 3 6
```



```
Output
/tmp/PRAA7NS8s9.o
5
2 4 9 88 4
2 100
2 4 100 88 4
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 12	PROGRAMS USING FILES

Question 1:

Write a program to create a file with integer values and open the file in read mode to read the content of the file then sort the content of the file in ascending order.

Rule:

The input file name should be named as "sample.txt".

Input format:

Given the input as a file containing integer values.

First line of the input consists of the number of values to be entered.

Second line of input consists of the values that to be sorted

Output format

Output consists of the sorted values

Aim:

To write the integer array in the document and sort it in ascending order.

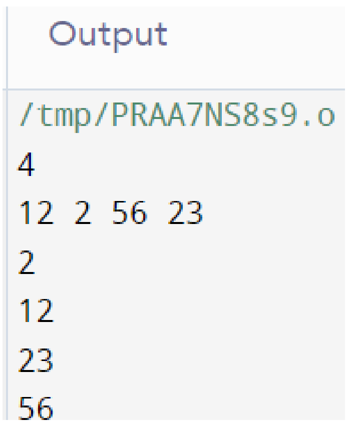
Algorithm:

1. The first input gives the size of the integer array and the array elements are read using a for loop.
2. Another for loop is used to write the array elements in the array.
3. Then the elements in the array are sorted within the document using nested for loop.
4. The ascended array is displayed in the output finally.

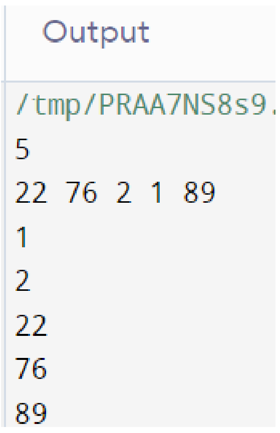
Program:

```
#include <iostream>
#include <fstream>
using std::ofstream;
using namespace std;

int main()
{
ofstream fout;
fout.open("sort.txt");
int n,i,j,t;
cin>>n;
int a[n];
for(i=0;i<n;i++)
{
    cin>>a[i];
}
for(i=0;i<n;i++)
{
    fout<<a[i];
}
fout.close();
ifstream file;
file.open("sort.txt");
int num;
while(file>>num)
{
    a[i]>>num;
    i++;
}
for(int i=0;i<n;i++)
{
    for(int j=i+1;j<n;j++)
    {
        if(a[i]>a[j])
        {
          t=a[i];
          a[i]=a[j];
          a[j]=t;
        }
    }
}
for(int i=0;i<n;i++)
{
    cout<<a[i]<<endl;
}
file.close();
    return 0;
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
4
12 2 56 23
2
12
23
56
```



```
Output
/tmp/PRAA7NS8s9.
5
22 76 2 1 89
1
2
22
76
89
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 13	PROGRAMS USING STL & LAMBDA FUNCTION

Question 1:

Write a code that adds two arrays and stores it in a third array of the same size using the transform function of STL.

Input Format

Three lines of input. First-line consists of a single integer which is the size of the three arrays. The second and third line consists of the elements of the 2 input arrays.

Output Format

The output displays the contents of the third array in a single line separated by a space. If constraints are violated, prints -1.

Constraints

0<Size of the array<=10

Aim:

To adds two arrays and stores it in a third array of the same size using the transform function of STL.

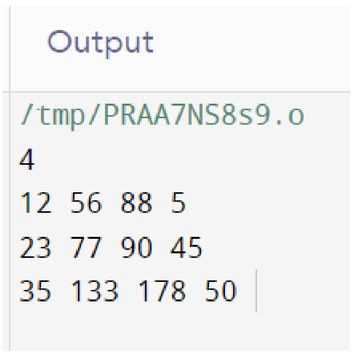
Algorithm:

1. First input is the size of two arrays and two for loops are used to read the array elements of two different arrays.
2. -1 is displayed if the number of array elements are given as negative.
3. Another for loop is used to add the elements at the same index of two arrays and stored in result array.
4. Finally, the result array is displayed in the output.

Program:

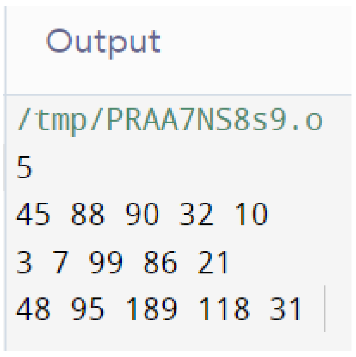
```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin>>n;
    if(n<1 || n>10)
    {
        cout<<"-1";
        return 0;
    }
    int arr1[n], arr2[n];
    int res[n];
    for(int i = 0;i<n;i++)
      cin>>arr1[i];
    for(int i =0;i<n;i++)
      cin>>arr2[i];
    // Single line code to add arr1[] and arr2[] and
    // store result in res[]
    transform(arr1, arr1+n, arr2, res, plus<int>());

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

Sample Output:



Output

```
/tmp/PRAA7NS8s9.o
4
12 56 88 5
23 77 90 45
35 133 178 50
```



Output

```
/tmp/PRAA7NS8s9.o
5
45 88 90 32 10
3 7 99 86 21
48 95 189 118 31
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 2:

Using the sort algorithm of STL, write a program that sorts a user-defined character array in ascending order.

Input Format

Two lines of input. First-line contains the number of elements. and the second line contains the elements of the array.

Output Format

Output is displayed as shown in sample test cases.

Aim:

To write a program that sorts a user-defined character array in ascending order.

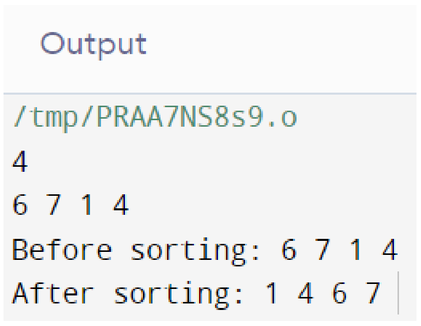
Algorithm:

1. The number of elements in the array is got as a input and for loop is used to read the array elements in the main function.
2. Another for loop is used to show the unsorted array.
3. Show function is called to sort and display the sorted array .

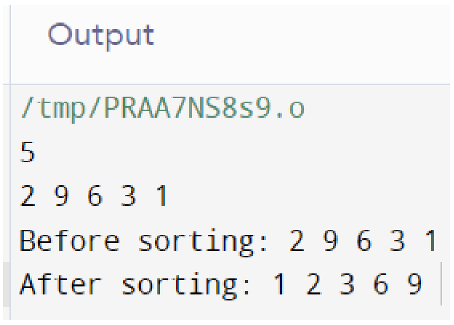
Program:

```
#include <iostream>
#include <algorithm>
using namespace std;
void show(char a[],int n)
{
    for(int i = 0; i < n; ++i)
        cout << a[i] << " ";
}
int main()
{
    int n;
    cin>>n;
    char a[n];
    for(int i=0;i<n;i++)
        cin>>a[i];
    cout << "Before sorting: ";
    show(a,n);
    sort(a, a+n);
    cout << "\nAfter sorting: ";
    show(a,n);
    return 0;
}
```

Sample Output:



```
/tmp/PRAA7NS8s9.o
4
6 7 1 4
Before sorting: 6 7 1 4
After sorting: 1 4 6 7
```



```
/tmp/PRAA7NS8s9.o
5
2 9 6 3 1
Before sorting: 2 9 6 3 1
After sorting: 1 2 3 6 9
```

Result:

Thus the program is run successfully and the outputs are verified successfully.

Question 3:

Write a program to create a list of integer elements and print the even elements in them using lambda expressions.

Input Format

The first line of the input consists of the value of n.

The next line of input is the list of elements.

Output Format

The output prints the even elements in the list/no even elements in the list.

Aim:

To create a list of integer elements and print the even elements in them using lambda expressions.

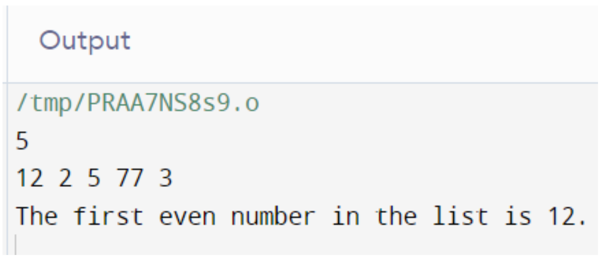
Algorithm:

1. Create a list of integer elements named numbers.
2. First input is the number of elements in the list.
3. For loop is used to get the input and push it back to the list.
4. The first even number is printed using if condition.

Program:

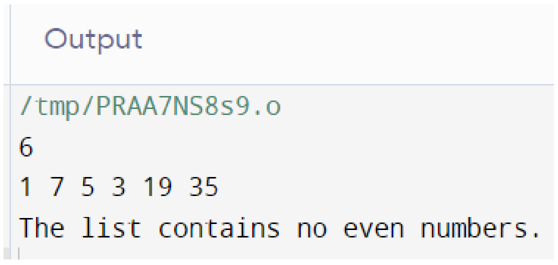
```
#include <list>
#include <algorithm>
#include <iostream>
using namespace std;
int main()
{
    list<int> numbers;
    int i,n,element;
    cin>>n;
    for(i=0;i<n;i++) {
        cin>>element;
        numbers.push_back(element);
    }
    const list<int>::const_iterator result =
        find_if(numbers.begin(), numbers.end(),[](int n) { return (n % 2) == 0; });
    if (result != numbers.end()) {
        cout << "The first even number in the list is " << *result << "." << endl;
    } else {
        cout << "The list contains no even numbers." << endl;
    }
}
```

Sample Output:



Output

```
/tmp/PRAA7NS8s9.o
5
12 2 5 77 3
The first even number in the list is 12.
```



Output

```
/tmp/PRAA7NS8s9.o
6
1 7 5 3 19 35
The list contains no even numbers.
```

Result

Thus the program is run successfully and the outputs are verified successfully.

Ex.No.: 14	**MINI PROJECT**

Write a Program to calculate the current bill.

Create a class currentBill with a virtual method double amount().

Create a class Fan that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

Create a class Light that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

Create a class TV that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

In the main method, prompt the user to enter the power rate of the appliance and the total hours used then create the necessary objects and call the methods.

Input Format

The first line consists of the power rating of the fan and the total hours used separated by space.

The second line consists of the power rating of Light and the total hours used separated by space.

The third line consists of the power rating of the TV and the total hours used separated by space.

Output Format

The output prints the bill amount.

Aim:

To calculate the current bill.

Algorithm:

1. parent class CurrentBill is created and virtual function amount is created.

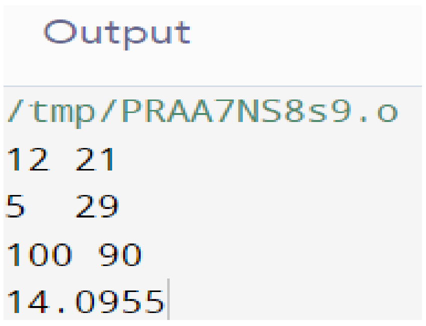
2. Three child classes are created namely Fan ,light and TV.
3. Each child class contains function amount () which differs in the calculation part.
4. Objects are created for each child class in the main function and values for that particular class are read and passed to the function.
5. Final output is the sum of the return values of amount() in all the classes.

Program:

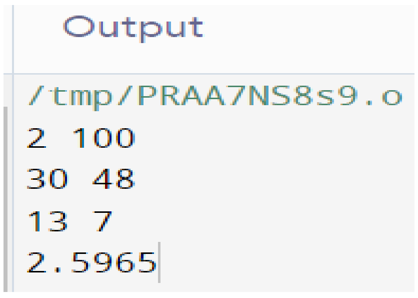
```
#include <iostream>
using namespace std;
class currentBill{
   public:
   virtual double amount()=0;
};

class Fan: public currentBill{
   public:
   int watts,hrs;
   double amount(){
      double t=watts*hrs;
      double a= (t/1000)*1.5;
      return a;
   }
};

class Light: public currentBill{
   public:
   int watts,hrs;
   double amount(){
      double t=watts*hrs;
      double a= (t/1000)*1.5;
      return a;
   }
};
```

```
class TV: public currentBill{
    public:
    int watts,hrs;
    double amount(){
        double t=watts*hrs;
        double a= (t/1000)*1.5;
        return a;
    }
};
int main()
{
   Fan f;
   cin>>f.watts>>f.hrs;
   Light l;
   cin>>l.watts>>l.hrs;
   TV t;
   cin>>t.watts>>t.hrs;
   cout<<f.amount()+l.amount()+t.amount();
   return 0;
}
```

Sample Output:



```
Output
/tmp/PRAA7NS8s9.o
12 21
5  29
100 90
14.0955
```



```
Output
/tmp/PRAA7NS8s9.o
2  100
30 48
13 7
2.5965
```

Result:

Thus the program is run successfully and the outputs are verified successfully.