**QUESTION-1**

**Program to find the count of even and odd numbers from a list of integers.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int a,n,e=0,o=0;

cout<<"Enter the Number of Elements in the List:";

cin>>n;

cout<<"Enter the Numbers:";

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

{

cin>>a;

if(a%2==0)

e++;

else

o++;

}

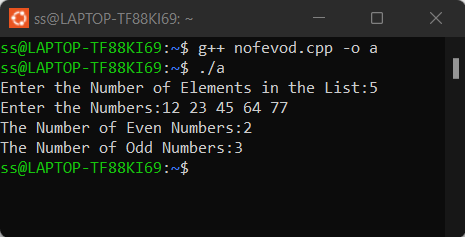
cout<<"The Number of Even Numbers:"<<e<<"\n";

cout<<"The Number of Odd Numbers:"<<o<<"\n";

return(0);

}

**OUTPUT:**



**QUESTION-2**

**Program to find the largest number among a list of numbers.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int n,l,a;

cout<<"Enter the Number of Elements of the List:";

cin>>n;

cout<<"Enter the Elements";

cin>>l;

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

{

cin>>a;

if (a>l)

l=a;

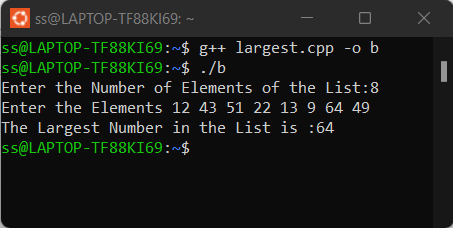
}

cout<<"The Largest Number in the List is :"<<l<<"\n";

return(0);

}

**OUTPUT:**



**QUESTION-3**

**Program to find the sum of find ‘n’ natural numbers.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int n,s=0;

cout<<"Enter the Number:";

cin>>n;

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

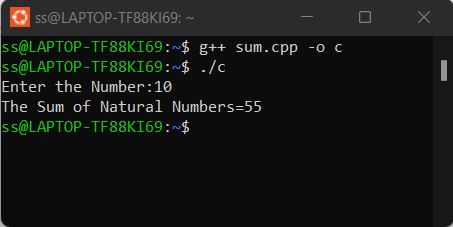
s=s+i;

cout<<"The Sum of Natural Numbers="<<s;

return(0);

}

**OUTPUT:**



**QUESTION-4**

**Program to find the sum of digits of an integer.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int n,s=0;

cout<<"Enter the Number:";

cin>>n;

if(n<0)

n=n\*-1;

while(n>0)

{

int r=n%10;

s=s+r;

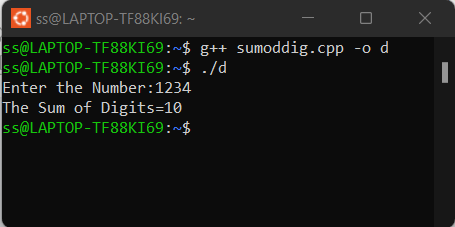
n=(n-r)/10;

}

cout<<"The Sum of Digits="<<s<<"\n";

return(0); }

**OUTPUT:**



**QUESTION 5**

**Program to reverse an integer.**

**PROGRAM*:***

#include<iostream>

using namespace std;

int main()

{

int n,s=0;

cout<<"Enter the Number:";

cin>>n;

while(n!=0)

{

int r=n%10;

s=s\*10+r;

n=(n-r)/10;

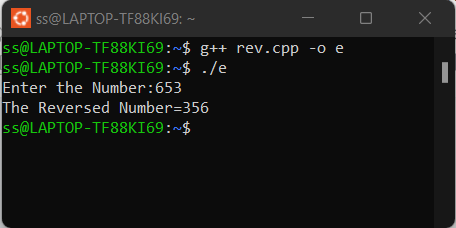
}

cout<<"The Reversed Number="<<s<<"\n";

return(0);

}

**OUTPUT*:***



**QUESTION 6**

**Program to accept limit and generate the pattern:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int l;

cout<<"Enter the Number of Rows:";

cin>>l;

for(int i=0;i<l;i++)

{

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

{

cout<<"\* ";

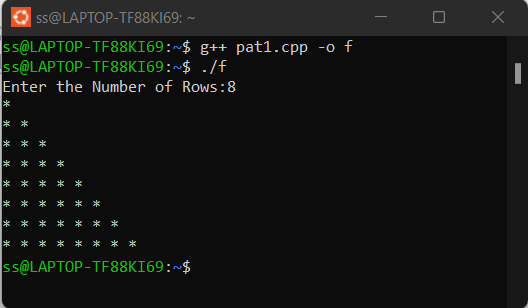
}

cout<<"\n";

}

}

**OUTPUT:**



**QUESTION 7**

**Program to accept limit and generate the pattern:**

\*

\* \* \*

\* \* \* \* \*

**PROGRAM:**#include<iostream>

using namespace std;

int main()

{

int l,a;

cout<<"Enter the Number of Rows:";

cin>>l;

a=l\*2;

for(int i=0;i<=l;i++)

{

for(int j=a;j>0;j--)

cout<<" ";

for(int k=0;k<(i\*2)-1;k++)

cout<<"\* ";

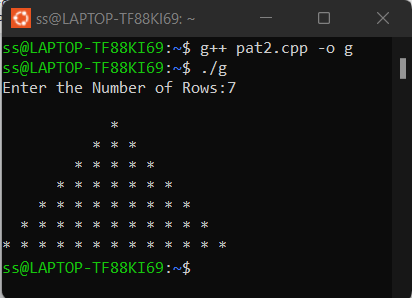
a=a-2;

cout<<"\n";

}

}

**OUTPUT:**



**QUESTION-8**

**Program to accept a limit and print the fibonocci series.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int a=0,b=1,c=0,l;

cout<<"Enter the Limit:";

cin>>l;

if(l<=0)

cout<<"Invalid Input";

if (l>0)

cout<<a<<" ";

if(l>1)

cout<<b<<" ";

for(int i=2;i<l;i++)

{

c=a+b;

cout<<c<<" ";

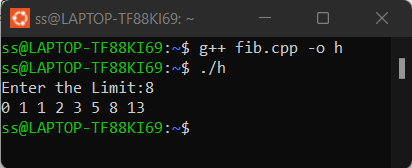
a=b;b=c;

}

return(0);

}

***OUTPUT:***



**QUESTION 9**

**Program to find the 2nd largest integer among a given set of integers.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int n,l=0,l1=0,a;

cout<<"Enter the Number of Elements in the List:";

cin>>n;

cout<<"Enter the Elements:";

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

{

cin>>a;

if (a>l)

{

l1=l;

l=a;

}

if((a>l1)&&(a<l))

l1=a;

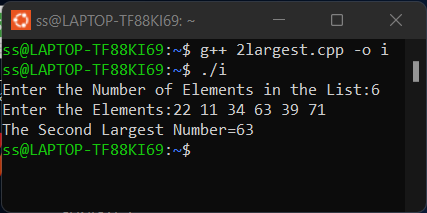
}

cout<<"The Second Largest Number="<<l1<<"\n";

return(0);

}

**OUTPUT :**



**QUESTION 10**

**Program to check whether the given integer is prime or not.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

int a,f=0;

cout<<"Enter the Number:";

cin>>a;

for(int i=2;i<a/2;i++)

{

if(a%i==0)

f++;

}

if(f!=0)

cout<<"The Number is Not Prime"<<"\n";

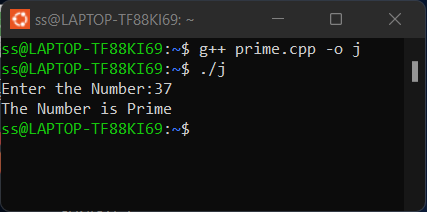
else

cout<<"The Number is Prime"<<"\n";

return(0);

}

**OUTPUT:**



**QUESTION-11**

**Program to check whether the number is Armstrong or not.**

**PROGRAM:**

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int n,s=0,a,b=0;

cout<<"Enter The Number:";

cin>>n;

a=n;

while(a>0)

{

int r=a%10;

b++;

a=(a-r)/10;

}

a=n;

while(a>0)

{

int r=a%10;

int k=pow(r,b);

s=s+k;

a=(a-r)/10;

}

if(s==n)

cout<<"The Number is Armstrong"<<"\n";

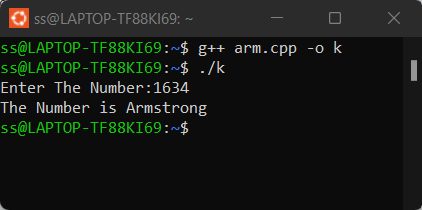
else

cout<<"The Number is not Armstrong"<<"\n";

return(0);

}

**OUTPUT :**



**QUESTION 12**

**Program to compute the value of nCr**

**PROGRAM :**

#include<iostream>

using namespace std;

int fact(int n)

{

        int f=1;

        while(n>0)

        {

                f=f\*n;

                n=n-1;

        }

        return (f);

}

int main()

{

        int n,r,s;

        cout<<"Enter the Values of n and r:";

        cin>>n;

        cin>>r;

        if(n==0)

{

                s=0;

cout<<"Value of nCr="<<s;

}

        else if (n>=r)

        {

                int a=fact(n);

                int b=fact(r);

                int c=fact(n-r);

                s=a/(b\*c);

                cout<<"Value of nCr="<<s;

         }

        else cout<<"Invalid Input";

        return(0);

}

**OUTPUT:**

**QUESTION 13**

**Program to generate the following pascals triangle**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**PROGRAM:**

#include<iostream>

using namespace std;

int fact(int n)

{

        int f=1;

        while(n>0)

        {

                f=f\*n;

                n=n-1;

        }

        return (f);

}

int ncr(int n, int r)

{

        if(n==0)

                return 0;

        int a=fact(n);

        int b=fact(r);

        int c=fact(n-r);

        int s=a/(b\*c);

        return(s);

}

int main()

{

        int l;

        cout<<"Enter the number of rows:" ;

        cin>>l;

        for(int i =0;i<l;i++)

        {

                for(int j=0;j<=i;j++)

                {

                        int s= ncr(i,j);

                        cout<<s<<" ";

                }

                cout<<"\n";

        }

}

**OUTPUT:**

**QUESTION 14**

**Program to generate the following pascals triangle**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**PROGRAM :**

#include<iostream>

using namespace std;

int fact(int n)

{

        int f=1;

        while(n>0)

        {

                f=f\*n;

                n=n-1;

        }

        return (f);

}

int ncr(int n, int r)

{

        if(n==0)

                return 0;

        int a=fact(n);

        int b=fact(r);

        int c=fact(n-r);

        int s=a/(b\*c);

        return(s);

}

int main()

{

        int l;

        cout<<"Enter the number of rows:" ;

        cin>>l;

        int z = l;

        for(int i =0;i<l;i++)

        {

                for(int k=z;k>0;k--)

                  cout<<" ";

                for(int j=0;j<=i;j++)

                {

                        int s= ncr(i,j);

                        cout<<s<<" ";

                }

                cout<<"\n";

                z=z-1;

        }

}

**OUTPUT:**

**QUESTION 15**

**Program to check whether the given year is a leap year or not.**

**PROGRAM:**

#include<iostream>

using namespace std;

int main()

{

    int year,k=0;

    cout<<"Enter the Year:";

    cin>>year;

    if(year%400==0)

        k=1;

    else if(year%100==0)

        k=0;

    else if(k%4==0)

        k=1;

    if(k==1)

        cout<<"The year is a Leap Year."<<"\n";

        else

            cout<<"The year is not a Leap Year."<<"\n";

}

**OUTPUT:**

**QUESTION 16**

**Program to check the roots of the given quadratic equation.**

**PROGRAM:**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    double a, b, c;

    cout << "Enter the coefficients a, b, and c: ";

    cin >> a >> b >> c;

    double d = b\*b - 4\*a\*c;

    if (d > 0) {

        double root1 = (-b + sqrt(d)) / (2\*a);

        double root2 = (-b - sqrt(d)) / (2\*a);

        cout << "Root 1: " << root1 << endl;

        cout << "Root 2: " << root2 << endl;

    }

    else if (d == 0) {

        double root = -b / (2\*a);

        cout << "Root: " << root << endl;

    }

    else {

        cout << "No real roots." << endl;

    }

    return 0;

}

**OUTPUT*:***