Q1. Write a program that

i. computes and returns the sum of all the integers between first and last

inclusive**.**

**PROGRAM**

#include<iostream>

using namespace std;

int add(int f, int l)

{

    if (f>l)

        {

            int temp;

            temp=f;

            f=l;

            l=temp;

        }

int sum=0;

    for(int i=f;i<=l;i++)

    {

        sum=sum+i;

    }

    return sum;

}

int main()

{   int first, last;

    cout<<"Enter the first ";

    cin>>first;

    cout<<"Enter the last ";

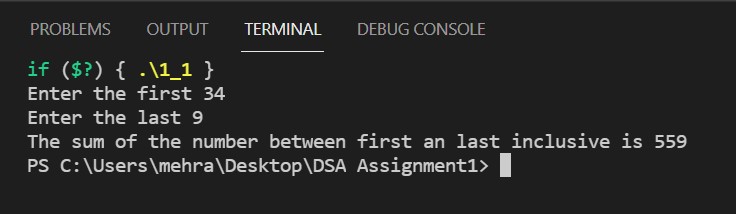
    cin>>last;

cout<<"The sum of the number between first an last inclusive is "<<add(first,last);

    return 0;

}

**OUTPUT**



ii. computes and returns the smallest positive integer n for which 1+2+3+...+n equals

or exceeds the value of &quot;goal&quot;.

**PROGRAM**

#include<iostream>

using namespace std;

int goal(int g)

{

    int sum=0,n;

    for(int i=0;;i++)

    {

        sum=sum+i;

        n=i;

        if(sum>=g)

        break;

    }

    return n;

}

int main()

{

    int g;

    cout<<"Enter the value of the goal ";

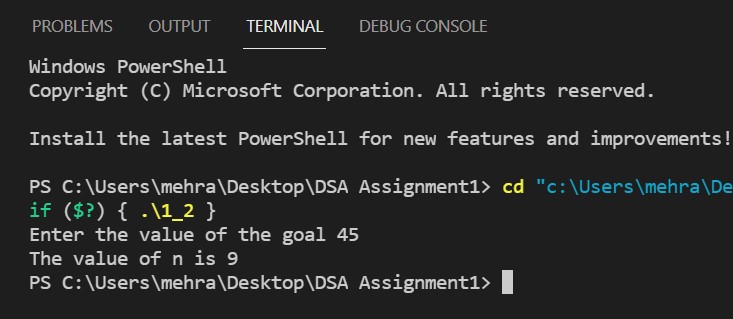
    cin>>g;

    cout<<"The value of n is "<<goal(g);

    return 0;

}

**OUTPUT**



iii. computes and returns the greatest common divisor (g.c.d.) of the arguments passed

to it.

**PROGRAM**

#include<iostream>

using namespace std;

int find(int n1, int n2)

{   int gcd;

    if(n2>n1)

        {

            int temp;

            temp;n1;

            n1=n2;

            n2=temp;

        }

    for (int i = 1; i <=  n2; ++i) {

        if (n1 % i == 0 && n2 % i ==0) {

        gcd = i;

        }

    }

    return gcd;

}

int main()

{

    int n1, n2, gcd;

    cout<<"Enter the two numbers\n";

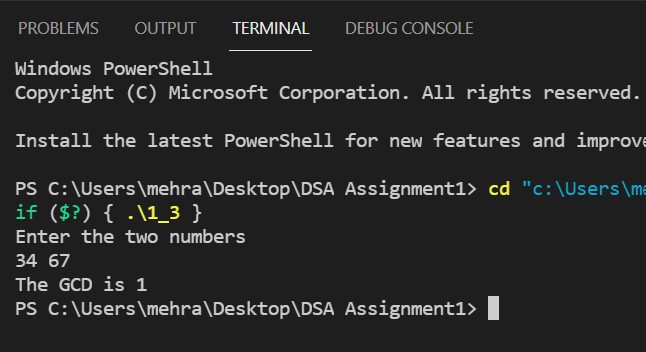
    cin>>n1>>n2;

cout<<"The GCD is "<<find(n1,n2);

    return 0;

}

**OUTPUT**



iv. Determines whether an integer is prime.

**PROGRAM**

#include<iostream>

using namespace std;

int main()

{

    int n,x;

    cout<<"Enter the integer ";

    cin>>n;

    if(n==1||n==0)

    {

        cout<<"Not a prime  number";

    }

    else{

        for(int i=2;i<=n/2;i++)

        {

            if(n%i==0)

            {

                x=0;

            }

        }

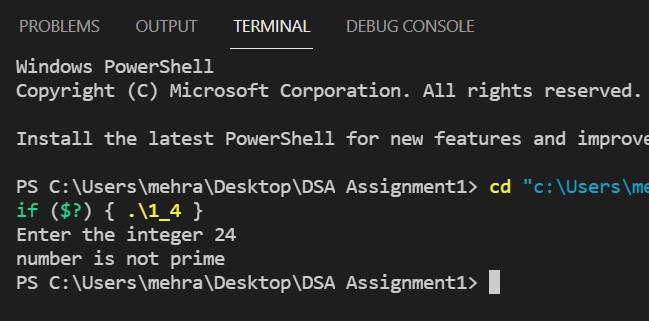
    }

    x?cout<<"Number is Prime":cout<<"number is not prime";

    return 0;

}

**OUTPUT**



v. Prints the English name of an integer from 1 to 9.

**PROGRAM**

#include<iostream>

#include<string>

using namespace std;

int main()

{

    string numbers[9] = { "one","two","three","four","five","six","seven","eight","nine"};

    int num;

    cout<<"Enter number between 1 to 9\n";

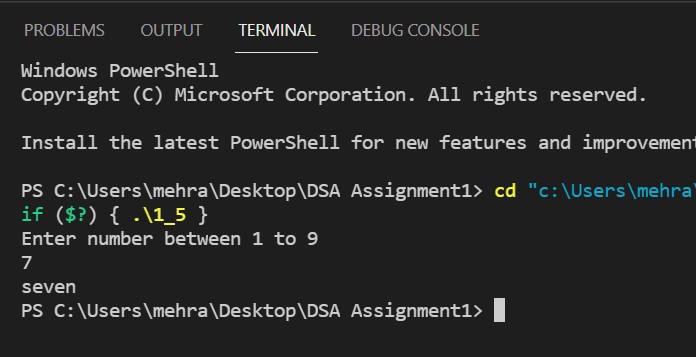
    cin>>num;

    cout<<numbers[num-1];

    return 0;

}

**OUTPUT**



vi. Reverses the order of the objects in an array.

**PROGRAM**

#include <iostream>

using namespace std;

void reverseArray(int arr[], int start, int end)

{

    while (start < end)

    {

        int temp = arr[start];

        arr[start] = arr[end];

        arr[end] = temp;

        start++;

        end--;

    }

}

void printArray(int arr[], int size)

{

   for (int i = 0; i < size; i++)

   cout << arr[i] << " ";

   cout << endl;

}

int main()

{

    int arr[] = {1, 2, 3, 4, 5, 6};

    int n = sizeof(arr) / sizeof(arr[0]);

    printArray(arr, n);

    reverseArray(arr, 0, n-1);

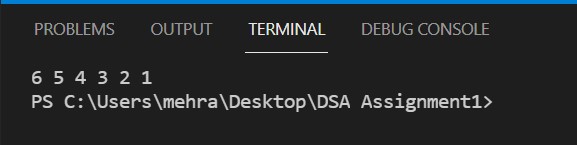
    cout << "Reversed array is" << endl;

    printArray(arr, n);

    return 0;

}

**OUTPUT**



vii. Finds the index of the largest number in an array.

**PROGRAM**

#include<iostream>

using namespace std;

int maxIndex(int arr[], int n)

{

    int max = arr[0];

    int i;

    for (i = 1; i < n; i++)

        {

            if (arr[i] > max)

           {max = arr[i];}

        }

    cout<<"Index of the maximum element of the array is "<<i<<endl;

    return max;

}

int main()

{

    int arr[] = {45,6,12,98,-3,5};

    int n = sizeof(arr)/sizeof(arr[0]);

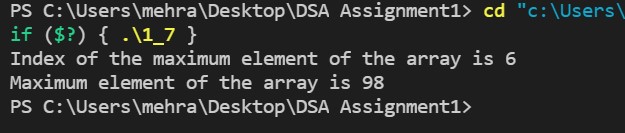
    int max=maxIndex(arr,n);

    cout<<"Maximum element of the array is "<<max;

    return 0;

}

**OUTPUT**



viii. Shifts the contents of array cells one cell to the right, with the last cell&#39;s contents

moved to the left end.

**PROGRAM**

#include<iostream>

using namespace std;

void Display(int x[],int n )

{

    for(int i =0; i<n;i++)

    {

        cout<<x[i]<<" ";//outputing array values.

    }

}

void shift(int x[],int n)

{

    int last = x[n-1];

    for(int i=n-2;i>=0;i--)

    {

        x[i+1]=x[i];

    }

    x[0]=last;

    cout<<"The shifted array is \n";

    Display(x,n);

}

int main()

{

    int arr[] = {12,94,-4,45,6,12,98,-3,5,78,-1};

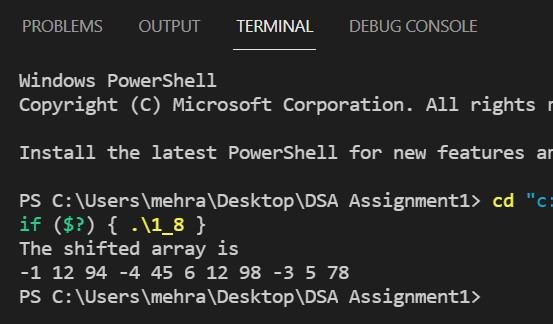
    int n = sizeof(arr)/sizeof(arr[0]);

    shift(arr,n);

    return 0;

}

**OUTPUT**



ix. Examines an array of integers and eliminates all duplication of values. The distinct

integers are all moved to the left part of the array.

**PROGRAM**

#include<iostream>

using namespace std;

void Display(int x[],int n )

{

    for(int i =0; i<n;i++)

    {

        cout<<x[i]<<" ";//outputing array values.

    }

}

int\* sort(int x[],int n)

{

    for(int i=0;i<n-1;i++)

    {

        for(int j=0;j<n;j++)

        {

            if(x[j]>x[j+1])

            {

                int temp;

                temp=x[j];

                x[j]=x[j+1];

                x[j+1]=temp;

            }

        }

    }

    return x;

}

int removeDuplicates(int arr[], int n)

{

    if (n==0 || n==1)

        return n;

    int temp[n];

    int j = 0;

    for (int i=0; i<n-1; i++)

        {if (arr[i] != arr[i+1])

            temp[j++] = arr[i];

        }

    temp[j++] = arr[n-1];

    for (int i=0; i<j; i++)

       { arr[i] = temp[i];}

    return j;

}

int main()

{

    int arr[] = {11,8,8,2,3,12,1,1,8,8,8,8,8,1,1,19,21,2,2,11,11,3,3,19,21};

    int n = sizeof(arr)/sizeof(arr[0]);

    int\* sorted\_arr = sort(arr,n);

    n = removeDuplicates(sorted\_arr,n);

    for(int i=0;i<n;i++)

    {

        cout<<\*(sorted\_arr+i)<<" ";

    }

    return 0;

}

**OUTPUT**



x. Copies numbers from two arrays into a third array. The numbers from the second

array are placed to the right of the numbers copied from the first array.

**PROGRAM**

#include<iostream>

using namespace std;

int\* merge(int x[],int y[],int arr3[],int n1,int n2,int n3)

{

    for(int i=0;i<n1;i++)

    {

        arr3[i]=x[i];

    }

    for(int i=n1,j=0;i<(n3);i++,j++)

    {

        arr3[i]=y[j];

    }

    return arr3;

}

int main()

{

    int arr1[]={88,109};

    int n1=sizeof(arr1)/sizeof(arr1[0]);

    int arr2[]={6,7,8,12,111,123591,12,7};

    int n2=sizeof(arr2)/sizeof(arr2[0]);

    int n3=n2+n1;

    int arr3[n3];

    int\* merged\_arr = merge(arr1,arr2,arr3,n1,n2,n3);

    cout<<"Merged array is \n";

    for(int i=0;i<n3;i++)

    {

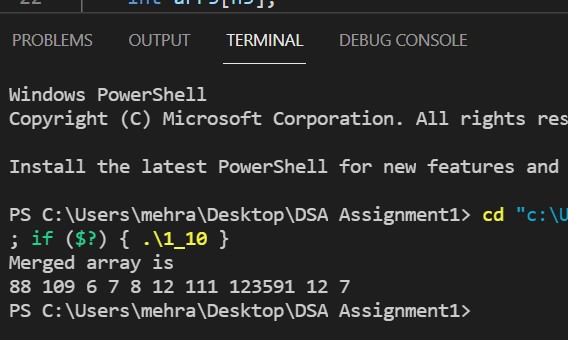
        cout<<\*(merged\_arr+i)<<" ";

    }

    return 0;

}

**OUTPUT**



Q2. Write a program to find sum of all elements of an array; write a program to find maximum of

elements of an array; write a program for linear search of an array.

**PROGRAM**

#include<iostream>

using namespace std;

int sum(int x[],int n)

{

    int sum=0;

    for(int i=0;i<n;i++)

    {

        sum=sum+x[i];

    }

    return sum;

}

int maxIndex(int arr[], int n)

{

    int max = arr[0];

    int i;

    for (i = 1; i < n; i++)

        {  if (arr[i] > max)

           {max = arr[i];}

        }

    cout<<"Index of the maximum element of the array is "<<i<<endl;

    return max;

}

int linearSearch(int x[],int n,int n1)

{

    for(int i=0;i<n;i++)

    {

        if(n1==x[i])

        {

            cout<<"Element found and is at index no. "<<i;

            return 1;

            break;

        }

    }

    return 0;

}

int main()

{

    int arr[] = {12,78,11,90,119,43};

    int n = sizeof(arr)/sizeof(arr[0]);

    cout<<"Sum of the elements of the array is "<<sum(arr,n)<<endl;

    int max=maxIndex(arr,n);

    cout<<"Maximum element of the array is "<<max<<endl;

    int e;

    cout<<"Enter the element you want to find \n";

    cin>>e;

    int check = linearSearch(arr,n,e);

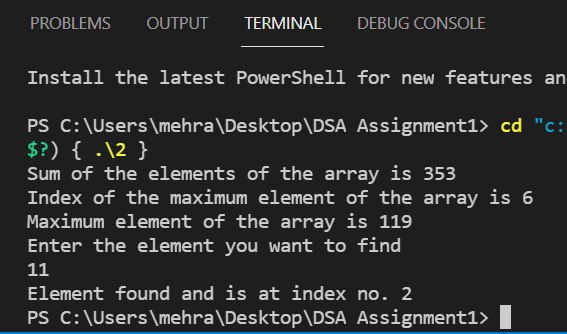
    if(!check)

    cout<<"Element not found\n";

return 0;

}

**OUTPUT**



Q3. Write a program for adding and subtracting Matrices using 2D Arrays.

**PROGRAM**

#include<iostream>

using namespace std;

int main()

{

    int r, c, a[100][100],b[100][100],sum[100][100];

    cout<<"Enter the number of rows \n";

    cin>>r;

    cout<<"Enter the number of columns \n";

    cin>>c;

    //ENTERING ELEMENTS THE 1ST MATRIX

    cout<<"Enter the elements of the first matrix \n";

    for(int i = 0; i < r; ++i)

      { for(int j = 0; j < c; ++j)

       {

           cout << "Enter element a" << i + 1 << j + 1 << " : ";

           cin >> a[i][j];

       }

      }

    //ENTERING ELEMENTS OF THE 2ND MATRIX

    cout<<"Enter the elements of the second matrix \n";

    for(int i = 0; i < r; ++i)

      { for(int j = 0; j < c; ++j)

       {

           cout << "Enter element a" << i + 1 << j + 1 << " : ";

           cin >> b[i][j];

       }

      }

    //ADDING THE MATRICES

    for(int i = 0; i < r; ++i)

        {

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

                {

                    sum[i][j] = a[i][j] + b[i][j];

                }

        }

    //SHOWING SUM OF THE MATRIX

    cout << endl << "Sum of two matrix is: " << endl;

    for(int i = 0; i < r; ++i)

    {    for(int j = 0; j < c; ++j)

        {

            cout << sum[i][j] << "  ";

            if(j == c - 1)

                cout << endl;

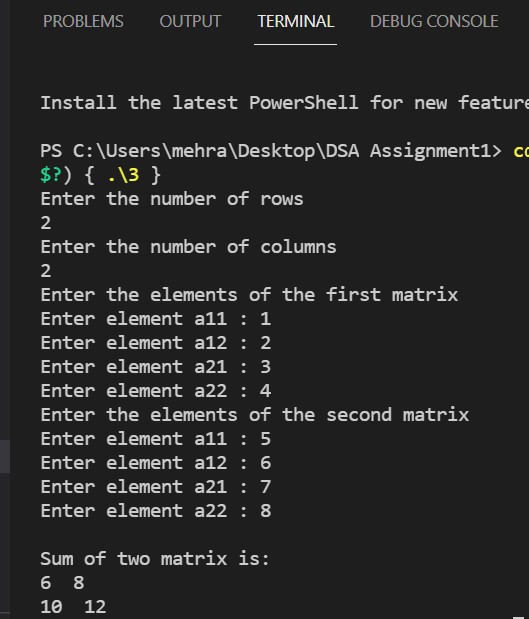
        }

    }

    return 0;

}

**OUTPUT**



Q4. Write a program to Multiply Matrices of 10X10.

**PROGRAM**

/\*MULTIPLICATION OF MATRICES\*/

#include<iostream>

using namespace std;

int main()

{

    int a[10][10], b[10][10], mult[10][10]={0}, r1, c1, r2, c2;

    cout << "Enter rows and columns for first matrix: ";

    cin >> r1 >> c1;

    cout << "Enter rows and columns for second matrix: ";

    cin >> r2 >> c2;

    while (c1!=r2)

    {

        cout << "column of first matrix not equal to row of second.";

        exit(0);

    }

    cout << endl << "Enter elements of matrix 1:" << endl;

    for(int i = 0; i < r1; ++i)

      {  for(int j = 0; j < c1; ++j)

        {

            cin >> a[i][j];

        }

      }

    cout << endl << "Enter elements of matrix 2:" << endl;

    for(int i = 0; i < r2; ++i)

        for(int j = 0; j < c2; ++j)

        {

            cin >> b[i][j];

        }

    //multiplying the matrices

    for(int i = 0; i < r1; ++i)

        for(int j = 0; j < c2; ++j)

            for(int k = 0; k < c1; ++k)

            {

                mult[i][j] += a[i][k] \* b[k][j];

            }

    //displaying the multiplied matrix

    cout << endl << "Output Matrix: " << endl;

    for(int i = 0; i < r1; ++i)

    {

        for(int j = 0; j < c2; ++j)

            {

                cout << " " << mult[i][j];

                if(j == c2-1)

                    cout << endl;

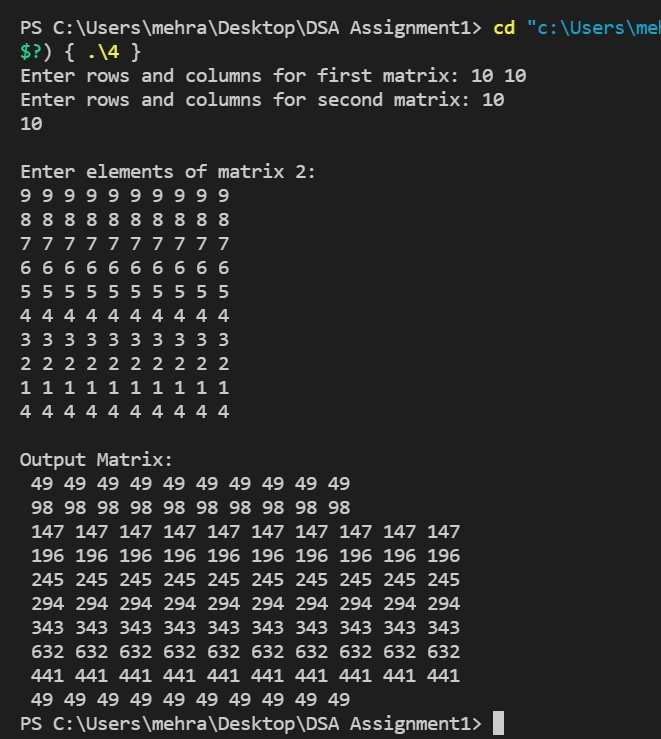
            }

    }

    return 0;

}

**OUTPUT**



Q5. Write a program for Linear Search using Functions.

**PROGRAM**

#include<iostream>

using namespace std;

int linearSearch(int x[],int n,int n1)

{

    for(int i=0;i<n;i++)

    {

        if(n1==x[i])

        {

            cout<<"Element found and is at index no. "<<i;

            return 1;

            break;

        }

    }

    return 0;

}

int main()

{

    int arr[] = {12,78,11,90,119,43};

    int n = sizeof(arr)/sizeof(arr[0]);

    int e;

    cout<<"Enter the element you want to find \n";

    cin>>e;

    int check = linearSearch(arr,n,e);

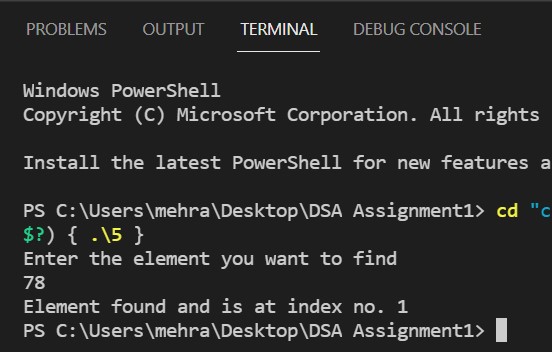
    if(!check)

    cout<<"Element not found\n";

return 0;

}

**OUTPUT**



Q6. Write a class for a rectangle. Create objects for this class and call the member functions to

find the area and perimeter of the rectangle.

**PROGRAM**

#include<iostream>

using namespace std;

class rectangle{

  public:

  rectangle(int l,int b)

  {

      length = l;

      breadth = b;

  }

    int length,breadth;

    int perimeter()

    {

        return 2\*(length+breadth);

    }

    int area()

    {

        return length\*breadth;

    }

};

int main()

{

    int x,y;

    cout<<"Enter the length and breadth of the rectangle\n";

    cin>>x>>y;

    rectangle r(x,y);

    cout<<"The perimeter of the rectangle is "<<r.perimeter()<<endl;

    cout<<"The area of the rectangle is "<<r.area()<<endl;

    return 0;

}

**OUTPUT**

