B.Sc. Hons. Computer Science

SEMESTER 1

Lab Record

Programming Fundamentals

HARSH BAMOTRA

Roll No. – AC-1216

TABLE OF CONTENTS

S No.	Program Details	Page No.
1	Program to compute the sum of first n terms of the given series.	2
2	Program to remove duplicates from an array.	4
3	Program to print the occurrences of each alphabet of a text entered as command line argument.	6
4	Menu driven program to perform different operations on string.	9
5	Program to merge two sorted arrays to get single sorted array.	16
6	Program for binary search:- (a)Using recursion (b)Without recursion	 18 21
7	Program to calculate GCD of two numbers :- (a)Using recursion (b)Without recursion	 24 27
8	Program to create a matrix class and perform matrix function .	28
9	Program to create a person class and inherit two classes from the base class and show polymorphism.	39
10	Program to create a triangle class and overload function and operators.	42
11	Program to read two numbers and perform division of the numbers and show divide by 0 error.	46
12	Rewrite the matrix class and then perform exception handling .	48
13	Program to create a student class and write the details of the students in a text file.	63
14	Program to copy the contents of a file to another after removing all whitespaces.	68

Practical-1

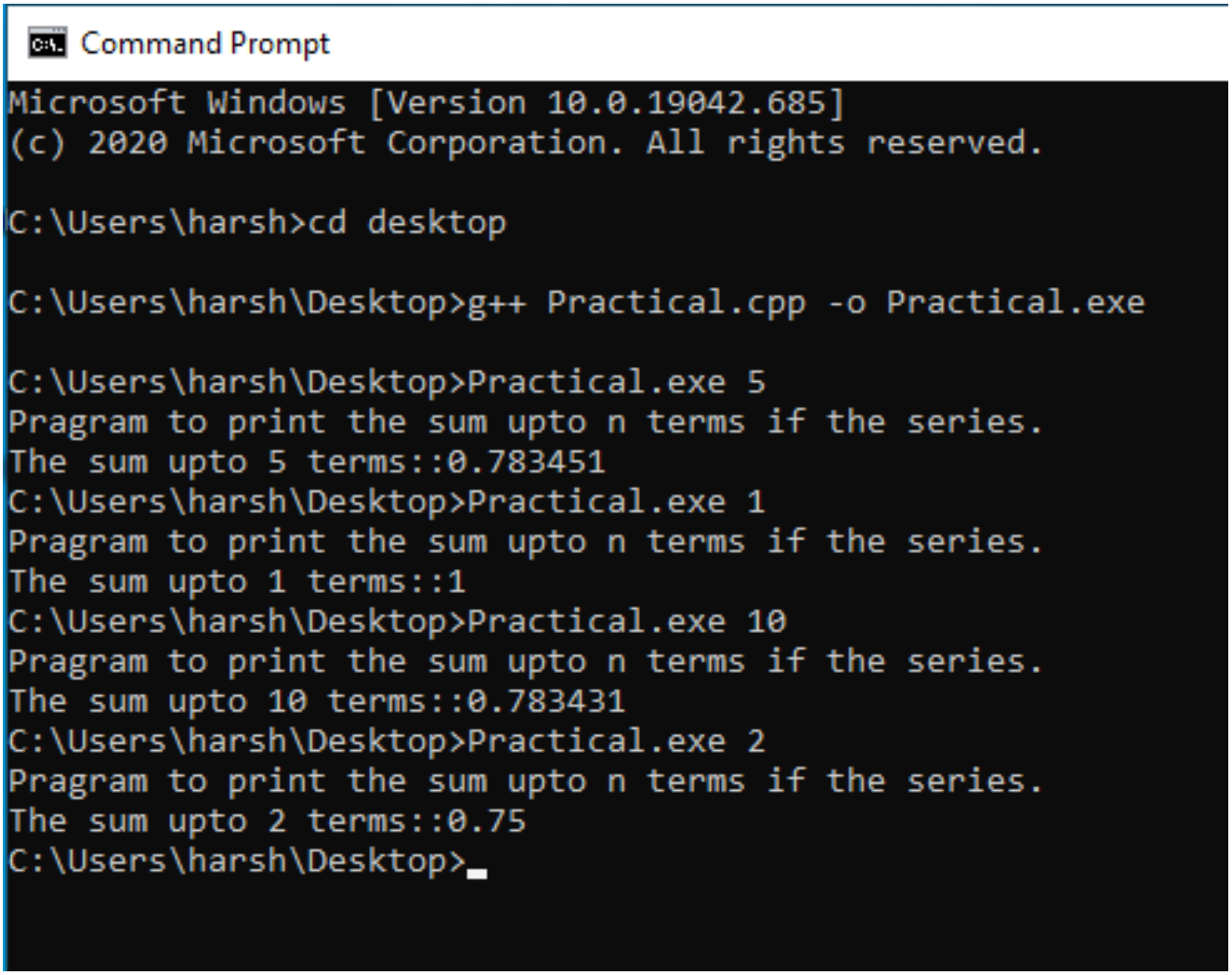
Q1. Program to compute the sum of the first n terms of the series by taking the number of terms in command line argument:-

S=1-1/(2^2)+1/(3^3)-……1/(n^n).

Code:-

```
//program to calculate the sum of first n terms
#include <iostream>
#include <math.h>
#include <cstdlib>

using namespace std;


int main(int argc , char *argv[])                    //taking input from the command line
{

   double y=0;                                       //declaring variable for calculating sum
   cout << "Pragram to print the sum upto n terms if the series." << endl;
   for(int i=1 ; i<atoi(argv[1])+1 ; i++)
        {

              if(i%2==0)
                    {

                          y-=1/pow(i,i);
                    }

              else                              //calculating the sum upto n terms
                    {

                          y+=1/pow(i,i);
                    }
        }


   cout << "The sum upto " << argv[1] << " terms::" << y;                 //printing the final result
   return 0;
}
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>g++ Practical.cpp -o Practical.exe
C:\Users\harsh\Desktop>Practical.exe 5
Pragram to print the sum upto n terms if the series.
The sum upto 5 terms::0.783451
C:\Users\harsh\Desktop>Practical.exe 1
Pragram to print the sum upto n terms if the series.
The sum upto 1 terms::1
C:\Users\harsh\Desktop>Practical.exe 10
Pragram to print the sum upto n terms if the series.
The sum upto 10 terms::0.783431
C:\Users\harsh\Desktop>Practical.exe 2
Pragram to print the sum upto n terms if the series.
The sum upto 2 terms::0.75
C:\Users\harsh\Desktop>

Practical-2

Q2. Program to remove duplicates from an array.

Code:-

```
//program to remove duplicates in an array
#include <iostream>
using namespace std;

int main()

{

  int n ,x=0;
  cout << "Enter the number of elements::";
  cin >> n;                              //taking array from the user
  int arr[n];
  for(int i=0 ;i<n ;i++)
        {
              cout << "Enter the elements::";
              cin >> arr[i];
        }

   cout << "The array you entered::";
   for(int i=0 ;i<n ;i++)
        {
              cout << arr[i] << " ";       //showing the inputed array to users
        }
    for(int i=0 ; i<n ;i++)
        {
              for(int j=i+1 ; j<n ;j++)
                    {
                          if(arr[i]==arr[j])                    //checking for duplicate elements
                                {
```

```
                                        for(int l=j ; l<n ; l++)
                                                {
                                                        arr[l]=arr[l+1];        //removing the duplicates
                                                }
                                        n--;                            //decreasing the length of the array
                                    }
                            }


        }


    cout << endl << "The array after deleting duplicates::";
    for(int i=0 ;i<n ;i++)
        {
            cout << arr[i] << " ";                  //printing the new updated array
        }


    return 0;


}
```

Output:-

Command Prompt

```
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\harsh>cd desktop

C:\Users\harsh\Desktop>cd Pracricales
The system cannot find the path specified.

C:\Users\harsh\Desktop>cd Practicales

C:\Users\harsh\Desktop\Practicales>g++ Practical-2.cpp -o Practical-2.exe

C:\Users\harsh\Desktop\Practicales>Practical-2.exe
Enter the number of elements::4
Enter the elements::1
Enter the elements::2
Enter the elements::2
Enter the elements::4
The array you entered::1 2 2 4
The array after deleting duplicates::1 2 4
C:\Users\harsh\Desktop\Practicales>
```

Practical-3

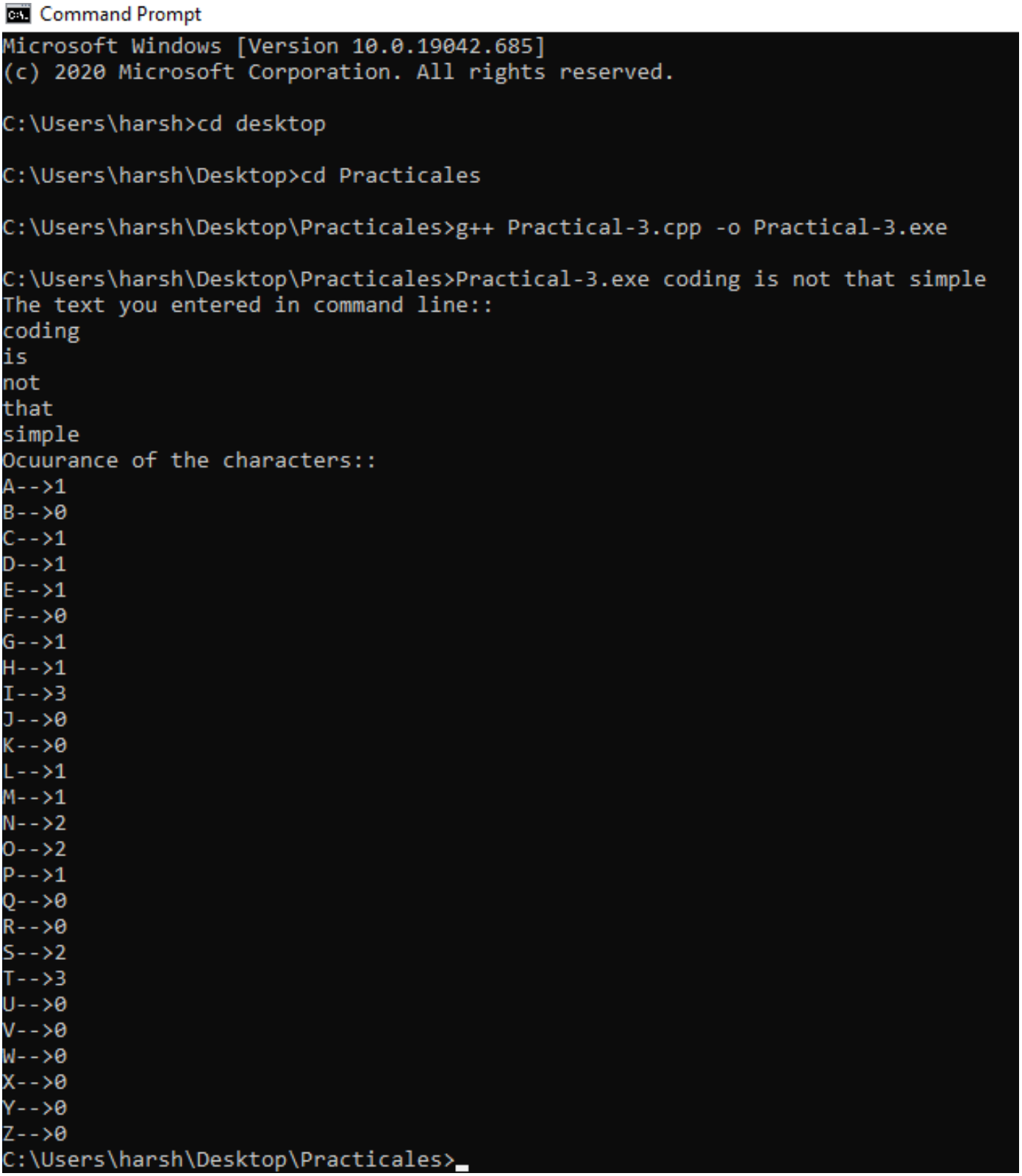
Q3. Program to print a table that shows the number of occurrence of alphabets in the text entered in the command line argument.

Code:-

```
//program to count the occurrence  of each character


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



int main(int argc, char *argv[])
   {

        string str="";                  //defining variables
        int count[26];
        int x;


        for(int i=0 ; i<26 ; i++)
                {

                        count[i]=0;             //initialising count to 0
                }



        cout << "The text you entered in command line::" << endl;
        for(int i=1 ; i<argc ; i++)
                {

                        cout << argv[i] << endl;
                        str+=argv[i];                           //adding the values of argv in str
                }
```

```
	for(int i=0 ; i<str.length() ; i++)
		{

			if(str[i]>='A' && str[i]<='Z')
				{

					x=int(str[i])-65;		//checking for alphabets

					count[x]++;			//counting the occurance

				}

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

					x=int(str[i])-97;		//checking for alphabets

					count[x]++;			//counting the occurrence

				}
		}




	cout<<"Occurrence of the characters::";

	for(int i=0 ; i<26 ; i++)

		{ 							//printing the final result
			cout << endl << char(65+i) << "-->" << count[i];

		}
	return 0;
}
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>cd Practicales
C:\Users\harsh\Desktop\Practicales>g++ Practical-3.cpp -o Practical-3.exe
C:\Users\harsh\Desktop\Practicales>Practical-3.exe coding is not that simple
The text you entered in command line::
coding
is
not
that
simple
Ocuurance of the characters::
A-->1
B-->0
C-->1
D-->1
E-->1
F-->0
G-->1
H-->1
I-->3
J-->0
K-->0
L-->1
M-->1
N-->2
O-->2
P-->1
Q-->0
R-->0
S-->2
T-->3
U-->0
V-->0
W-->0
X-->0
Y-->0
Z-->0
C:\Users\harsh\Desktop\Practicales>

Practicle-4

Q4. Menu driven program to perform different operation on strings without using inbuilt string function.

Code:-

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

int len(string s)
   {                                    //defining function to find the length of the string
      int lent=0;
      while(true)
         {
            if(s[lent]>char(31) && s[lent]<char(123))
                  {
                        lent+=1;
                  }
             else
                  {
                        break;
                  }
         }
      return lent;

   }


void adress(string s)
   {                                                          //defining function to find the address of string
      cout << "Adress of the characters::" << endl;
      for(int i=0 ; i<len(s) ; i++)
         {
            cout << s[i] << " -> " << (void*)&s[i] << endl;
         }
   }


string con(string s)
   {                                    //defining function to concatenate two strings
      string second , f;
      cout << "Enter the second string to Concatenate::";
      cin >> second;
      f=s+second;
      return f;
   }
```

```
void com(string s)
   {                                    //defining function to compare two strings
      string second;
      bool equal=true;
      cout << "Enter the second string to compare::";
      cin >> second;
      for(int i=0 ; i<len(s) && i<len(second) ; i++)
         {
            if(s[i]!=second[i])
               {
                  equal=false;
               }

         }
      if(equal==true)
          {
                    cout << "The strings are equal!!";
          }
       else
          {
                    cout << "The strings are not equal!!";
          }

   }

string rev(string s)
   {                            //defining function to find the reverse of the string
      char f[100];
      int l=len(s);
      for(int i=0 ; i<l ; i++)
         {
                    f[i]=s[l-1-i];
         }
      return f;
   }

string upper(string s)
   {                            //defining function to convert the lower case to upper
      int v;
      for(int i=0 ; i<len(s) ; i++)
         {
            if(s[i]>=char(97) && s[i]<=char(122))
                    {
                              v=int(s[i]);
                              s[i]=char(v-32);
                    }
            else
                    {
                              continue;
                    }
         }
      return s;
```

```
}

int main()

{

   string x;       //defining variables
   int y;
   cout << "Enter the string::";
   getline(cin , x);                                  //showing the menu to user
   cout << "*****************************************" << endl;
   cout << "1.Adress of character" << endl << "2.Concatenate" << endl << "3.Compare two strings" << endl;
   cout << "4.Length of string" << endl << "5.Lower case to Upper case" << endl << "6.Reverse" << endl;
   cout << "*****************************************" << endl;
   cout << "Enter your choice (1,2,3,4,5 or 6) ::";
   cin >>y;
   switch(y)
         {                                          //creating switch case and printing the final result
                  case 1:
                           adress(x);
                           break;
                  case 2:
                           cout << "The string after concatenate::" << con(x);
                           break;
                  case 3:
                           com(x);
                           break;
                  case 4:
                           cout << "The lenght of the string is::" << len(x);
                           break;
                  case 5:
                           cout << "The string in upper case::" << upper(x);
                           break;
                  case 6:
                           cout << "The string after reverse::" << rev(x);
                           break;
                  default:
                           cout << "Wrong Input !!";
                           break;
         }
   return 0;

}
```

Output:-

Function-1

```
Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\harsh>cd desktop

C:\Users\harsh\Desktop>cd Practicales

C:\Users\harsh\Desktop\Practicales>g++ Practical-4.cpp -o Practical-4.exe

C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is great
*****************************************
1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
*****************************************
Enter your choice (1,2,3,4,5 or 6) ::1
Adress of the characters::
H -> 0x72fd00
a -> 0x72fd01
r -> 0x72fd02
s -> 0x72fd03
h -> 0x72fd04
  -> 0x72fd05
i -> 0x72fd06
s -> 0x72fd07
  -> 0x72fd08
g -> 0x72fd09
r -> 0x72fd0a
e -> 0x72fd0b
a -> 0x72fd0c
t -> 0x72fd0d
```

Function-2

```
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is
*********************************************
1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
*********************************************
Enter your choice (1,2,3,4,5 or 6) ::2
Enter the second string to Concatenate::great
The string after concatenate::Harsh is great
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
```

Function-3

```
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is great
************************************************

1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
************************************************

Enter your choice (1,2,3,4,5 or 6) ::3
Enter the second string to compare::Harsh is great
The strings are equal!!
```

Function-4

```
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is great
************************************************

1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
************************************************

Enter your choice (1,2,3,4,5 or 6) ::4
The lenght of the string is::14
```

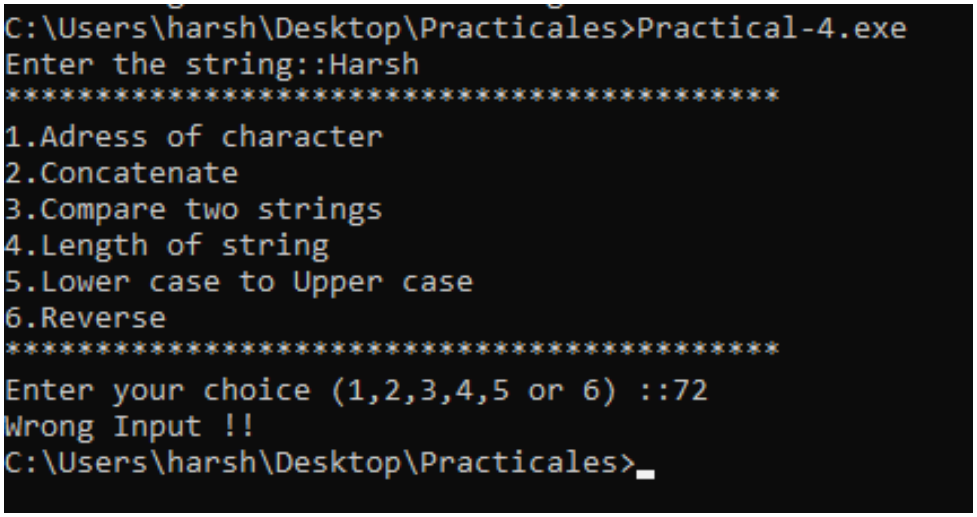
Function-5

```
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is great
*********************************************

1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
*********************************************

Enter your choice (1,2,3,4,5 or 6) ::5
The string in upper case::HARSH IS GREAT
C:\Users\harsh\Desktop\Practicales>
```

Function-6

```
C:\Users\harsh\Desktop\Practicales>Practical-4.exe
Enter the string::Harsh is great
*********************************************

1.Adress of character
2.Concatenate
3.Compare two strings
4.Length of string
5.Lower case to Upper case
6.Reverse
*********************************************

Enter your choice (1,2,3,4,5 or 6) ::6
The string after reverse::taerg si hsraH
```

Checking Exception



Practicle-5

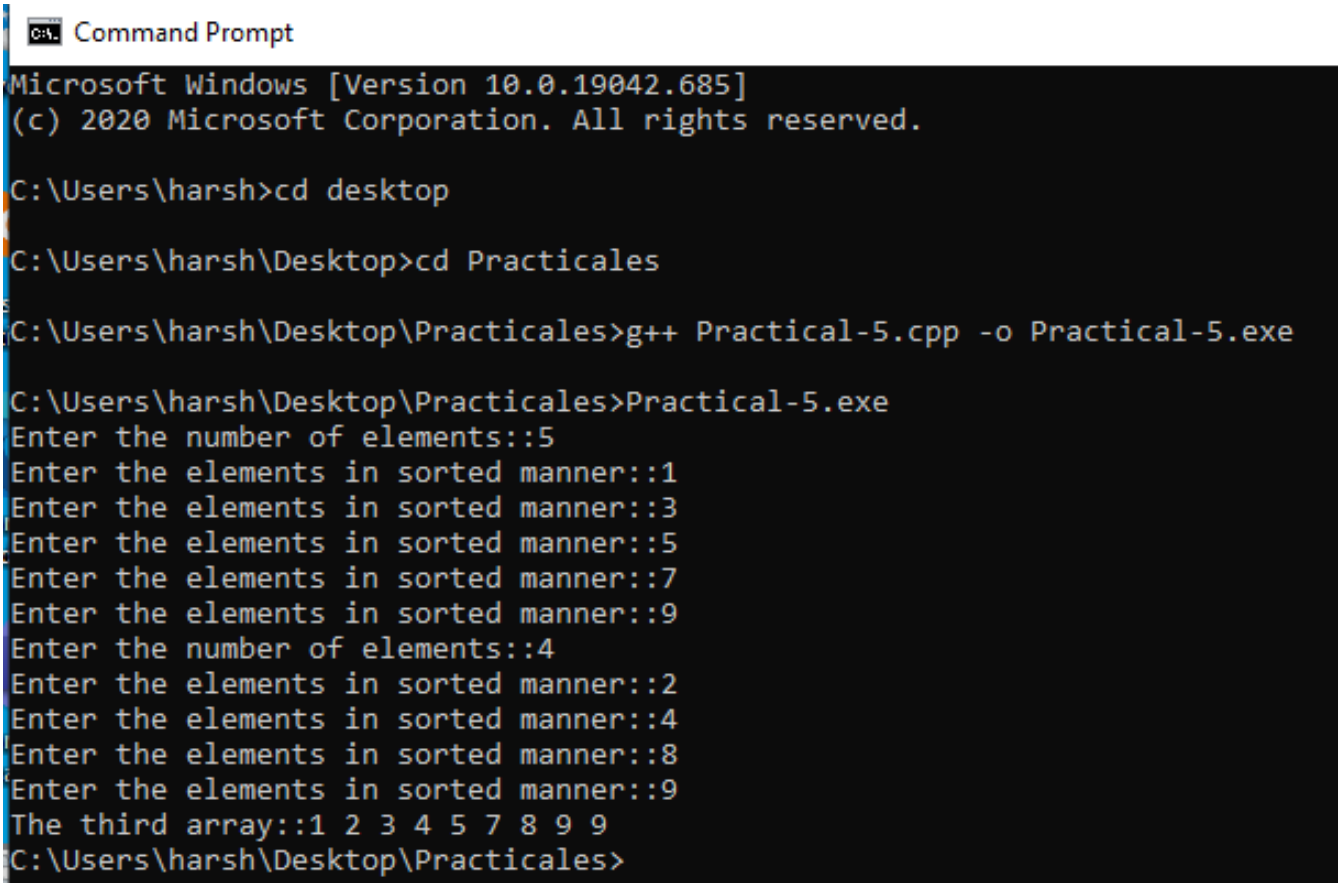
Q5. Program to merge two sorted array into one sorted array.

Code:-

```
//program to merge two sorted array
#include <iostream>
using namespace std;

int main()

{

   int n1 , n2 , n3;                                          //Declaring variables
   cout << "Enter the number of elements::";
   cin >> n1;
   int arr1[n1];
   for(int i=0 ; i<n1 ; i++)
         {
               cout << "Enter the elements in sorted manner::";
               cin >> arr1[i];                              //inputting first array
         }

   cout << "Enter the number of elements::";
   cin >> n2;
   int arr2[n2];
   for(int j=0 ; j<n2 ; j++)
         {
               cout << "Enter the elements in sorted manner::";          //inputing second array
               cin >> arr2[j];
         }

   n3=n1+n2;
   int arr3[n3], x=0 , y=0 ,k=0;
   while(x<n1 && y<n2)
         {
               if(arr1[x]<=arr2[y])
                     {
                           arr3[k++]=arr1[x++];
                     }
               else                                          //comparing the elements and
                     {                                      //copying them in the third array
                           arr3[k++]=arr2[y++];
                     }
         }
    while(x<n1)
         {
               arr3[k++]=arr1[x++];
         }
```

```
    while(y<n2)                         //copying the remaining elements
        {
            arr3[k++]=arr2[y++];
        }


   cout << "The third array::";
   for(int p=0; p<n3; p++)
        {
            cout << arr3[p] << " ";        //printing the final merged array
        }

   return 0;
}
```

Output:-



Practical-6(a)

Q. Program for binary search using recursion.

Code:-

```
//Harsh Bamotra AC-1216
//Program for binary search using recursion


#include <iostream>

using namespace std;


//defining function for binary search
int binary(int s , int high , int low , int arr[])
  {
        if(high>=1)                                     //checking for empty array
                {


                        int mid=low+(high-1)/2;
                        if(arr[mid]==s)                         //searching in the mid index
                                {                               //and if found then returning the index

                                        return mid;

                                }
                        else if(arr[mid]>s)                     //checking in the lower part
                                {                               //of the array


                                        return binary(s , mid-1 , low , arr );
                                }


                        else                                    //checking in the upper part
                                {                               //of the array


                                        return binary(s , high , mid+1 , arr);
                                }
                }

        return -1;                                      //returning -1 to check for exception
  }
```

```
int main()

    {
        int n , s;                                          //defining variables
        cout << "Enter the number of elements::";
        cin >> n;                                          //taking number of elements
        int arr[n];
        cout << "Enter the elements in ascending order::" << endl;

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

            {
                cin >> arr[i];                             //initializing the elements in the array
            }
        cout << "The array you entered::";

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

            {
                cout << arr[i] << " ";                     //printing the array
            }
        cout << endl << "Enter the element you want to search::";
        cin >> s;                                          //taking the search element
        int r=binary(s , n-1 , 0 , arr);

        if(r==-1)

            {
                cout << "Element not found !!";
            }
        else                                               //printing the final result
            {
                cout << "Element found at index::" << r;
            }
        return 0;
  }
```

Output:-

Normal case

```
C:\Users\harsh\Desktop>recur_binary.exe
Enter the number of elements::6
Enter the elements in ascending order::
1
3
5
7
9
12
The array you entered::1 3 5 7 9 12
Enter the element you want to search::12
Element found at index::5
C:\Users\harsh\Desktop>
```

Exception case

```
C:\Users\harsh\Desktop>recur_binary.exe
Enter the number of elements::5
Enter the elements in ascending order::
1
3
4
7
54
The array you entered::1 3 4 7 54
Enter the element you want to search::2
Element not found !!
C:\Users\harsh\Desktop>
```

Practical-6(b)

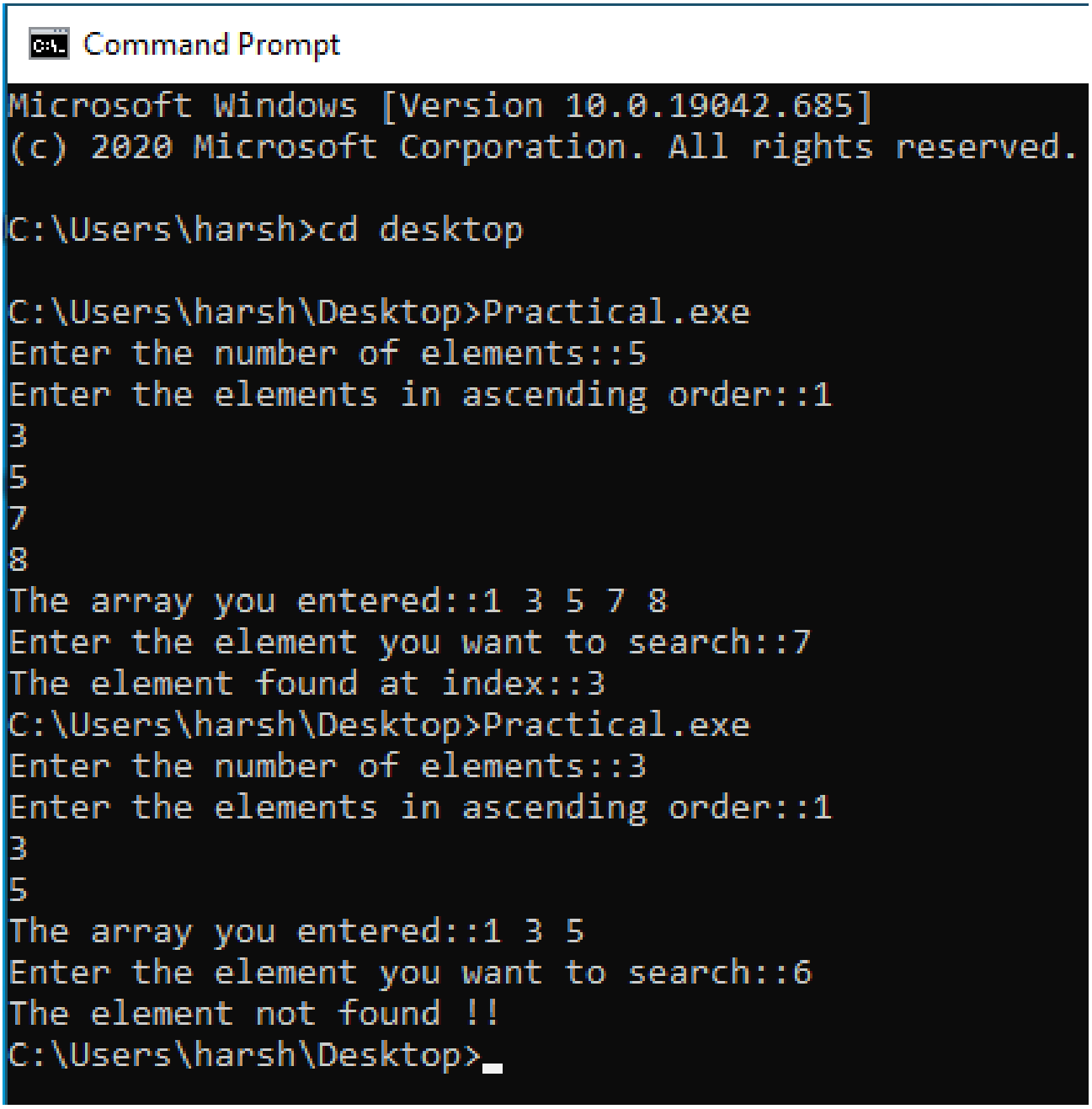
Q6. Program to binary search without using recursion.

Code:-

```
//program for binary search using function
#include <iostream>
using namespace std;


void binary_search(int n , int s , int arr[])              //defining function
   {
        int high=n , low=0 , i=0 , mid;                  //defining variables
        bool flag=false;                                  //defining bool to check exception
        while(high>low)
                {
                        mid=(high+low)/2;
                        if(arr[mid]==s)                   //searching in the mid position
                                {
                                        flag=true;
                                        break;
                                }
                        else if(s<arr[mid])
                                {
                                        high=mid;
                                }                         //searching in the lower part of the array
                         else
                                {
                                        low=mid+1;
                                }                           //searching in the upper part of the array

                }
```

```
    if(flag==false)                                          //checking for exeption
        {

            cout << "The element not found !!";
        }

    else                                                     //printing the final result
        {

            cout << "The element found at index::" << mid;
        }
}


int main()
{

    int n , s;
    cout << "Enter the number of elements::";

    cin >> n;

    int arr[n];
    cout << "Enter the elements in ascending order::";

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

        {

            cin >> arr[i];              //taking array from the user
        }

    cout << "The array you entered::";

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

        {
            cout << arr[i] << " ";          //printing the array
        }

    cout << endl << "Enter the element you want to search::";
    cin >> s;                               //taking the element to search

    binary_search(n , s , arr);             //using the function defined earlier

    return 0;

}
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>Practical.exe
Enter the number of elements::5
Enter the elements in ascending order::1
3
5
7
8
The array you entered::1 3 5 7 8
Enter the element you want to search::7
The element found at index::3
C:\Users\harsh\Desktop>Practical.exe
Enter the number of elements::3
Enter the elements in ascending order::1
3
5
The array you entered::1 3 5
Enter the element you want to search::6
The element not found !!
C:\Users\harsh\Desktop>

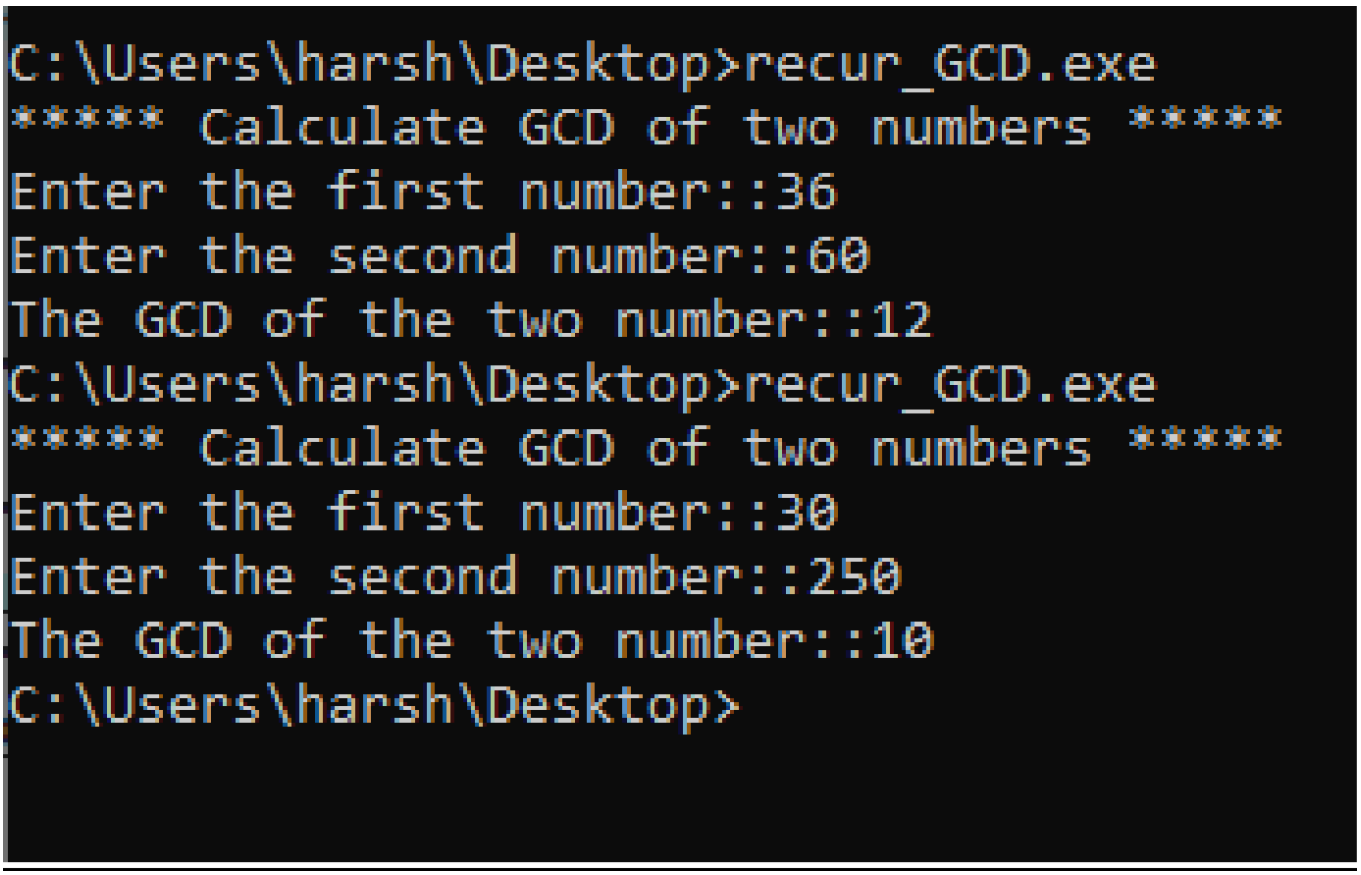
Practical-7(a)

Q. Program to calculate the GCD of two numbers using recursion.

Code:-

```
//Harsh Bamotra AC-1216
//Program to calculate GCD of two numbers using recursion

#include <iostream>
using namespace std;

//defining function to calculate GCD
int recur_GCD(int n1 , int n2)
  {
        if(n1==0 || n2==0)                          //checking if one of the number is 0
                {
                        return 0;                   //exception case
                }
        else if(n1==n2)                                 //if n1 is equal to n2
                {
                        return n1;                  // exit case
                }
        else if(n1>n2)                                                  //checking if n1 is greater than n2
                {
                        return recur_GCD(n1-n2 , n2);
                }
        else                                                            //if n2 is greater than n1
                {
                        return recur_GCD(n1 , n2-n1);
                }
  }

int main()
  {
        cout << "***** Calculate GCD of two numbers *****" << endl;
        int n1 , n2;                                                    //defining variables
        cout << "Enter the first number::";
        cin >> n1;                                                      //taking first number
        cout << "Enter the second number::";
        cin >> n2;                                                      //taking second number

        cout << "The GCD of the two number::" << recur_GCD(n1 , n2);        //printing the final result
        return 0;
  }
```

Output:-


C:\Users\harsh\Desktop>recur_GCD.exe
***** Calculate GCD of two numbers *****
Enter the first number::36
Enter the second number::60
The GCD of the two number::12
C:\Users\harsh\Desktop>recur_GCD.exe
***** Calculate GCD of two numbers *****
Enter the first number::30
Enter the second number::250
The GCD of the two number::10
C:\Users\harsh\Desktop>

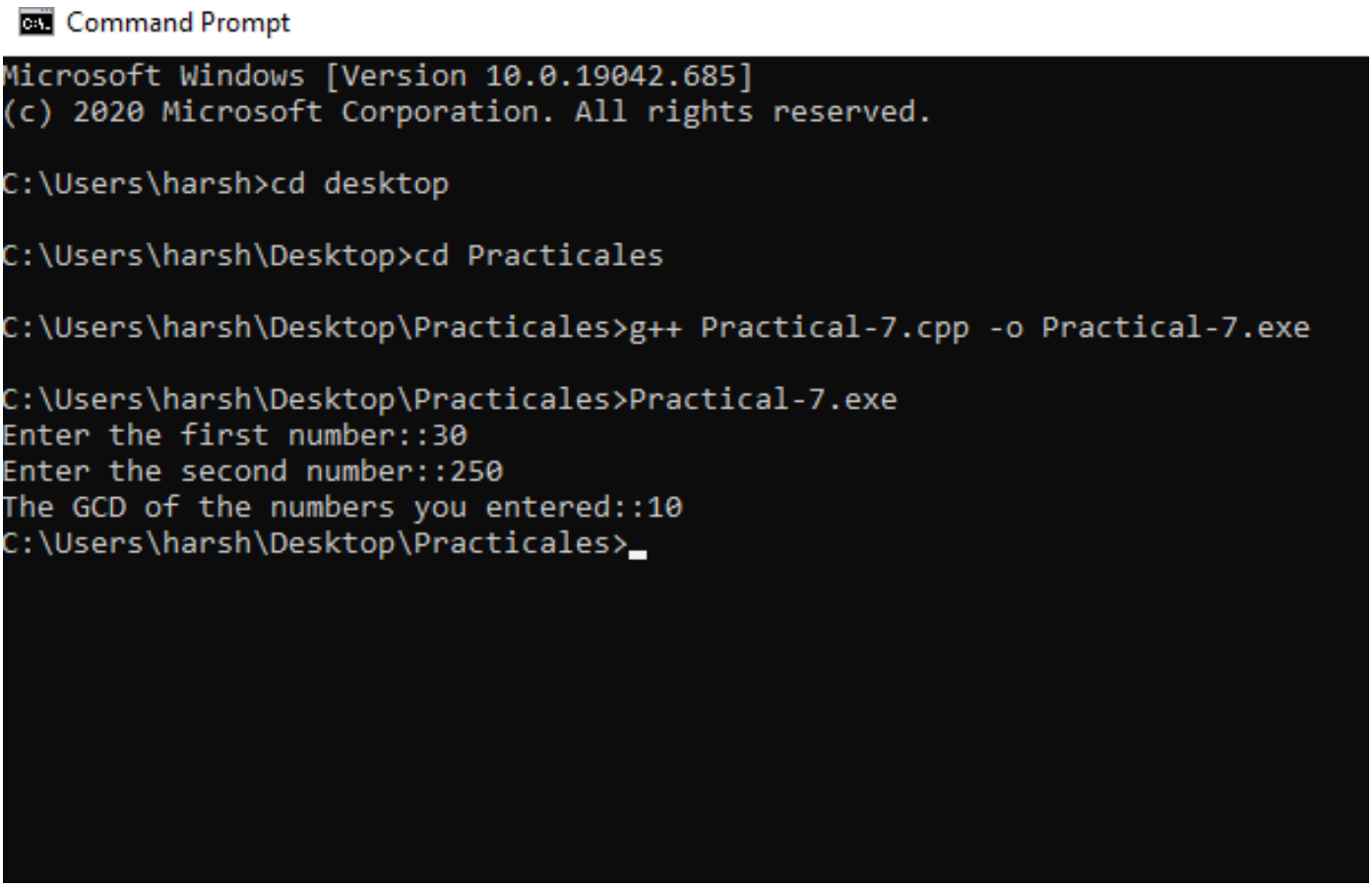
Practical-7(b)

Q7. Program to calculate the GCD of two numbers without using recursion.

Code:-

```
//program to calculate GCD of two numbers
#include <iostream>

using namespace std;


int gcd(int n1 , int n2)        //defining function
  {
        int gc;
        for(int i=1 ; i<=n1 && i<=n2 ; i++)
                {
                        if(n1%i==0 && n2%i==0)                  //checking for the common factors
                                {
                                        gc=i;      //storing the common factors in gc
                                }
                }
        return gc;                                      //returning gcd of the numbers
  }


int main()


  {
        int n1 , n2 ;                                           //defining variables
        cout << "Enter the first number::";

        cin >> n1;

        cout << "Enter the second number::";                    //taking the numbers from the user

        cin >> n2;

        cout << "The GCD of the numbers you entered::" << gcd(n1 , n2);       //printing the final result

        return 0;

  }
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>cd Practicales
C:\Users\harsh\Desktop\Practicales>g++ Practical-7.cpp -o Practical-7.exe
C:\Users\harsh\Desktop\Practicales>Practical-7.exe
Enter the first number::30
Enter the second number::250
The GCD of the numbers you entered::10
C:\Users\harsh\Desktop\Practicales>

Practical-8

Q. Program to create a matrix class and perform basic matrix functions .

Code:-

```
//Harsh Bamotra AC-1216
//Program to perform matrix function using matrix class

#include <iostream>
using namespace std;

//************************* Defining matrix class ************************//

class matrix
  {
        private:
        int row , col;                          //defining private members

        public:
        int arr[10][10];                         //defining public members
        void setData(int n1 , int n2)           //defining function to take input in the private members
                {
                        row=n1;                 //initializing private members
                        col=n2;
                }

        void create_arr()                                       //defining function to create a matrix by taking matrix elements
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << "Enter the elements at index [" << i << "][" << j << "]::";
                                                        cin >> arr[i][j];
                                                }
                                }
                }

        void display_arr()                                      //defining function to print the elements of the matrix
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        cout << endl;
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << arr[i][j] << " ";
                                                }
                                }
                }
```

```
        void trans();                           //defining function trans() for transpose of the matrix

//*********************overloading operators to perform basic matrix functions *********************//

        matrix operator +(matrix m)                             //overloading + operator to perform sum of matrix
                {
                        matrix temp;                    //defining a matrix class object for storing the result
                        temp.row=m.row;                 //initializing the private members of temp
                        temp.col=m.col;
                        if(row==m.row && col==m.col)                    //checking the order of the two matrix
                                {                                       //if they are equal or not
                                for(int i=0 ; i<row ; i++)
                                        {
                                                for(int j=0 ; j<col ; j++)              //adding the elements
                                                        {                               // and initializing them in arr of temp
                                                                temp.arr[i][j]=arr[i][j]+m.arr[i][j];
                                                        }
                                        }
                                }
                        else
                                {
                                        cout << "Error !! The order of the matrix are not same. ";
                                }                                                       //handling exception

                        return temp;                    //returning the temp
                }



        matrix operator -(matrix m)                     //overloading – operator for subtracting two matrix
                {
                        matrix temp;                      //defining a matrix class object for storing the result
                        temp.row=m.row;                   //initializing the private members of temp
                        temp.col=m.col;
                        if(row==m.row && col==m.col)            //checking if the order of the matrix is same or not
                                {
                                for(int i=0 ; i<row ; i++)
                                        {
                                                for(int j=0 ; j<col ; j++)              //subtracting the elements and
                                                        {                               //initializing them in the arr of temp
                                                                temp.arr[i][j]=arr[i][j]-m.arr[i][j];
                                                        }
                                        }
                                }
                        else
                                {
                                        cout << "Error !! The order of the matrix are not same. ";
                                }                                                               //handling exeption

                        return temp;                    //returning temp
                }
```

```
        matrix operator *(matrix m)                     //overloading * operator for multiplying two matrix
                {
                        matrix temp;                    //defining a matrix class object for storing the result
                        temp.row=row;                   //initializing the private members of temp
                        temp.col=m.col;
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {                               // initializing the whole matrix to 0
                                                        temp.arr[i][j]=0;
                                                }
                                }
                        if(col==m.row)          //checking is the col of first matrix is equal to the row of the second matrix
                                {
                                        for(int i=0 ; i<row ; i++)                      //multiplying the elements and
                                                {                                       //initializing them in arr of temp
                                                        for(int j=0 ; j<m.col ; j++)
                                                                {
                                                                        for(int k=0 ; k<col ; k++)
                                                                                {
                                                                                        temp.arr[i][j]+=arr[i][k]*m.arr[k][j];
                                                                                }
                                                                }
                                                }

                                }
                        else
                                {
                                        cout << "Error !! the column of the fisrt matrix not equal to the row of second.";
                                }                                       //handling exception
                        return temp;
                }                                       //returning temp

   };

void matrix :: trans()                                  //defining the logic for transpose function
        {
                int arr1[10][10];                       //defining a matrix
                for(int i=0 ; i<row ; i++)
                        {
                                for(int j=0 ; j<col ; j++)
                                        {
                                                arr1[i][j]=arr[j][i];           //transposing the elements and initializing
                                        }                                       // in the arr1
                        }

                cout << "The matrix after transpose::";
                for(int i=0 ; i<row ; i++)
                        {
                                cout << endl;
                                for(int j=0 ; j<col ; j++)
                                        {
                                                cout << arr1[i][j] << " ";              //printing the transposed matrix
```

```
                                        }
                                }
        }


int main()
  {
        int r1 , r2 , c1 , c2 , x , y;                          //defining variables
        matrix m1 , m2 , m3 ;                                   //defining matrix class objects
        cout << "*****Enter the details of first matrix*****" << endl << endl;
        cout << "Enter the number of row::";
        cin >> r1;                                              //taking number of rows from the user
        cout << "Enter the number of columns::";
        cin >> c1;                                              //taking number of columns from the user
        m1.setData(r1 , c1);
        m1.create_arr();                                        //taking elements of the matrix from the user
        cout << "The matrix you entered::" << endl;
        m1.display_arr();                                       //printing the matrix

        cout << endl << endl << "*****Enter the details of second matrix*****" << endl << endl;
        cout << "Enter the number of row::";
        cin >> r2;                                              //taking number of rows from the user
        cout << "Enter the number of columns::";
        cin >> c2;                                              //taking number of columns from the user
        m2.setData(r2 , c2);
        m2.create_arr();                                        //taking elements of the matrix from the user
        cout << "The matrix you entered::" << endl;
        m2.display_arr();                                       //printing the matrix

        //**********************Printing the menu to the user*********************************//

        cout << endl << "*********************************************************************" << endl;
        cout << "1.Sum" << endl << "2.Product" << endl << "3.Transpose" << endl << "4.Subracting" << endl ;
        cout << "*********************************************************************" << endl;
        cout << "Enter your choice(1 , 2 , 3 or 4)::";
        cin >> x;                                               //taking users choice
        if(x==1)
                {
                        m3=m1+m2;                               //adding the matrix
                        if(r1==r2 && c1==c2)
                                {
                                        cout << "The sum of the matrix::" << endl;
                                        m3.display_arr();                       //printing the result
                                }
                }
        else if(x==2)
                {
                        m3=m1*m2;                               //multiplying the matrix
                        if(c1==r2)
                                {
                                        cout << "The product of the matrix::" << endl;
                                        m3.display_arr();                       //printing the result
                                }
                }
```

```
	else if(x==3)
		{
			cout << "**********************************************************************" << endl;
			cout << "Which matrix you want to transpose (1 or 2)::";
			cin >> y;                                                    //taking users choice
			if(y==1)
			 {
				m1.trans();                                  //transposing the first matrix
			 }
			else if(y==2)
				{
				m2.trans();                                  //transposing the second  matrix
				}
			else
				{
				cout << "Wrong Input!!" << endl ;             //handling exception
				}
		}
	else if(x==4)
		{
			m3=m1-m2;                                                    //subtracting the matrix
			if(r1==r2 && c1==c2)
				{
				cout << "The subtraction of the matrix::";
				m3.display_arr();                                        //printing the result
				}
		}
	else
		{
			cout << "Wrong input !!!";                                   //handling exception
		}


	return 0 ;
}
```

Output:-

1. Sum of two matrix

Select Command Prompt

```
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\harsh>cd desktop

C:\Users\harsh\Desktop>cd AC-1216

C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
The sum of the matrix::

2 4 6
2 4 6
2 4 6
C:\Users\harsh\Desktop\AC-1216>
```

2. Multiplying the matrix

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [0][2]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
Enter the elements at index [1][2]::1
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::1
Enter the elements at index [2][2]::1
The matrix you entered::

1 1 1
1 1 1
1 1 1

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::1
The matrix you entered::

1 1
1 1
1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
The product of the matrix::

3 3
3 3
3 3
C:\Users\harsh\Desktop\AC-1216>
```

3. Transpose of the matrix

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
**********************************************************************
Which matrix you want to transpose (1 or 2)::1
The matrix after transpose::
1 1 1
2 2 2
3 3 3
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
```

4. Subtraction of two matrix

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::3
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::3
Enter the elements at index [1][1]::1
Enter the elements at index [2][0]::2
Enter the elements at index [2][1]::3
The matrix you entered::

1 2
3 1
2 3

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::3
Enter the elements at index [1][1]::1
Enter the elements at index [2][0]::2
Enter the elements at index [2][1]::3
The matrix you entered::

1 2
3 1
2 3
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::4
The subtraction of the matrix::
0 0
0 0
0 0
C:\Users\harsh\Desktop\AC-1216>
```

5. Handling different exceptions

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::1
Enter the number of columns::1
Enter the elements at index [0][0]::1
The matrix you entered::

1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::1
Enter the elements at index [0][0]::1
The matrix you entered::

1
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::4545
Wrong input !!!
```

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::1
Enter the elements at index [0][0]::1
Enter the elements at index [1][0]::1
The matrix you entered::

1
1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
Error !! The order of the matrix are not same.
C:\Users\harsh\Desktop\AC-1216>
```

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::1
Enter the elements at index [0][0]::1
The matrix you entered::

1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
Error !! the column of the fisrt matrix not equal to the row of second.
C:\Users\harsh\Desktop\AC-1216>
```

```
C:\Users\harsh\Desktop\AC-1216>Matrix.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::22
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::
2
Enter the elements at index [1][1]::2
The matrix you entered::

22 2
2 2

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::2
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::2
Enter the elements at index [1][1]::2
The matrix you entered::

2 2
2 2
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
**********************************************************************
Which matrix you want to transpose (1 or 2)::45
Wrong Input!!
```

Practical-9

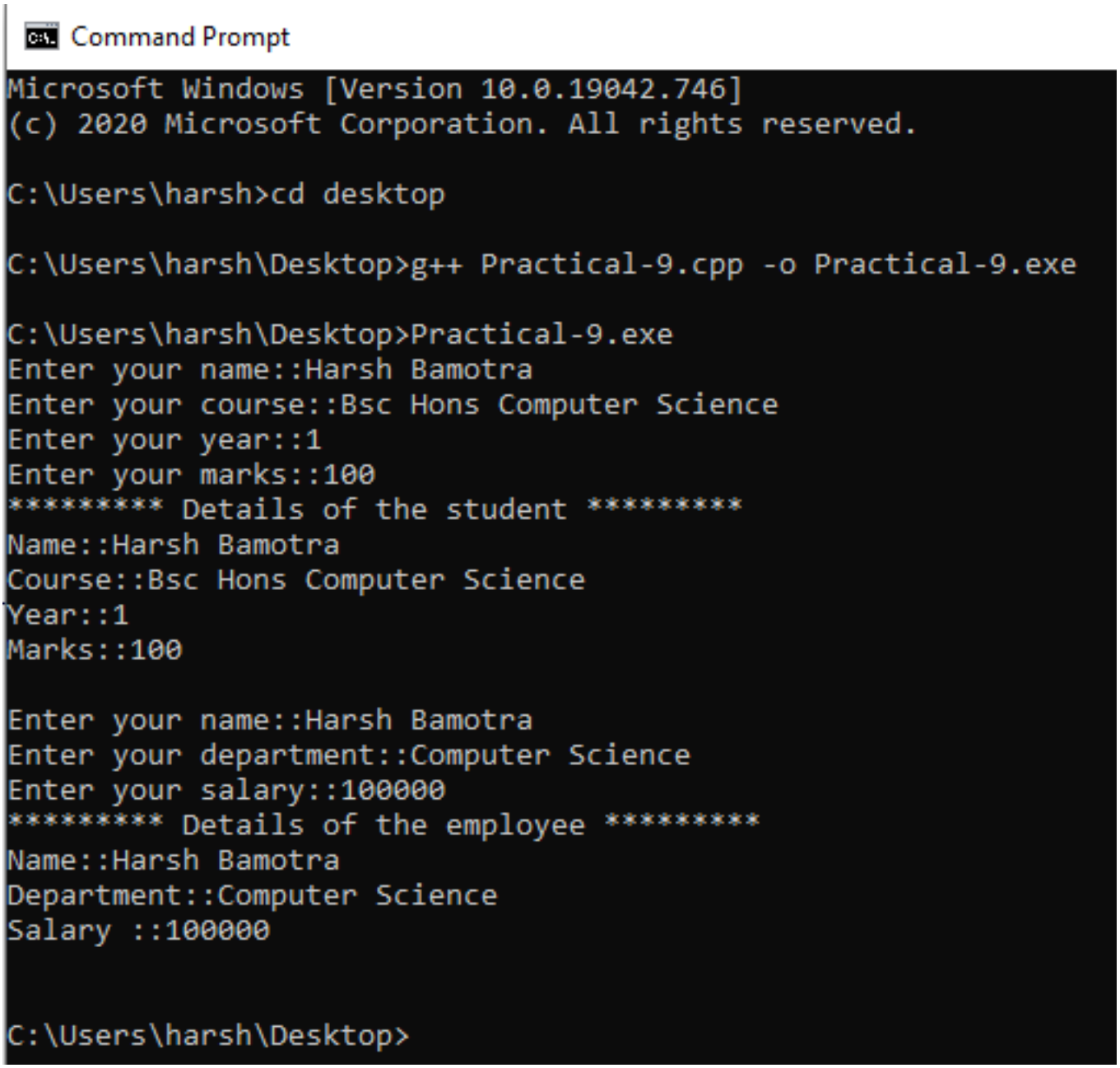
Q. Program to create a person class having names as data member in it and define two inherited classes Student and Employee .

Code:-

```
//Harsh Bamotra AC-1216
//Program to create a person class and show polymorphism

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

//defining class person
class person
  {
        public:                                         //defining public members
        string name;
        person()                                        //defining constructor
                {
                        cout << "Enter your name::";
                        getline(cin, name);
                        fflush;
                }
        virtual void display()                          //defining virtual function display to print the details
                {
                        cout << "Name::" << name;
                }
  };
//inheriting student class from person class
class student : public person
  {
        private:                                        //defining private members
        string course;
        int year , marks;

        public:                                         //defining public members
        student()                                       //defining constructor
                {
                        cout << "Enter your course::";
                        getline(cin , course);
                        fflush;
                        cout << "Enter your year::";
                        cin >> year;
                        cout << "Enter your marks::";
                        cin >> marks;
                }
```

```
        void display()                                          //overriding function display
                {
                        cout << "Name::" << name << endl;
                        cout << "Course::" << course << endl;
                        cout << "Year::" << year << endl;
                        cout << "Marks::" << marks << endl << endl;
                }
        };
//inheriting class employee from person
class employee : public person
  {
        private:                                                //defining private members
        string department;
        int salary;

        public:                                                  //defining pubic members
        employee()                                              //defining constructor
                {
                        cout << "Enter your department::";
                        getline(cin , department);
                        fflush;
                        cout << "Enter your salary::";
                        cin >> salary;
                }

        void display()                                          //overriding function display
                {
                        cout << "Name::" << name << endl;
                        cout << "Department::" << department << endl;
                        cout << "Salary ::" << salary << endl << endl;
                }
  };
int main()
  {
        person *p;                                                  //defining class pointer

        student obj1;                                               //defining class object
        p=&obj1;                                                    //pointing the p pointer to object of student
        cout << "********* Details of the student *********" << endl;
        p -> display();                                         //displaying the data
        cin.ignore();

        employee obj2;                                              //defining class object
        p=&obj2;                                                    //pointing the p pointer to object of employee
        cout << "********* Details of the employee *********" << endl;
        p -> display();                                             //displaying the data
        cin.ignore();

        return 0;
  }
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>g++ Practical-9.cpp -o Practical-9.exe
C:\Users\harsh\Desktop>Practical-9.exe
Enter your name::Harsh Bamotra
Enter your course::Bsc Hons Computer Science
Enter your year::1
Enter your marks::100
********** Details of the student **********
Name::Harsh Bamotra
Course::Bsc Hons Computer Science
Year::1
Marks::100
Enter your name::Harsh Bamotra
Enter your department::Computer Science
Enter your salary::100000
********** Details of the employee **********
Name::Harsh Bamotra
Department::Computer Science
Salary ::100000
C:\Users\harsh\Desktop>

Practical-10

Q. Program to create a triangle class and overloading area function and assignment and equality operators .

Code:-

```
//Harsh Bamotra AC-1216
//Program to create a triangle class

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

class triangle
  {                                  //defining class triangle
        private:                            //defining private members
        double base , height , side;

        public:
        void setData(double x , double y , double z)       //function to initialize the private members
                {
                        base=x;
                        height=y;
                        side=z;
                }

        void setData(double x , double y)          //overloading setData
                {
                        base=x;
                        height=y;
                        side=0;
                }

        double area(double base , double height)     //function to calculate the area
                {
                        double area;
                        area=0.5*base*height;
                        return area;
                }

        double area(double base , double height , double side)  //overloading area
                {
                        double area , sp;
                        sp=(base+height+side)/2;
                        area=sqrt(sp*(sp-base)*(sp-height)*(sp-side));
                        return area;
                }
```

```
        void operator =(triangle &m)              //overloading assignment operator
                {
                        m.base=base;
                        m.height=height;
                        m.side=side;
                }



        bool operator ==(triangle &m)          //overloading equality operator
                {
                        if(m.base==base && m.height==height && m.side==side)
                                {
                                        return true;
                                }
                        else
                                {
                                        return false;
                                }
                }
  };




int main()
  {
        double s1 , s2 , s3;              //defining variables
        int ch;
        triangle t1 , t2;                 //defining objects

        cout << endl << "***************** MENU ************************" << endl;
        cout << "1. Find area using three sides." << endl;
        cout << "2. Find area using two sides." << endl;
        cout << "Enter your choice ::";
        cin >> ch;

        switch (ch)
                {
                        case 1:
                                cout << endl << "***** Enter the sides of the triangle *****" << endl;
                                cout << "Enter the first side of the triangle::";
                                cin >> s1;
                                cout << "Enter the second side of the triangle::";         //taking the length of sides
                                cin >> s2;
                                cout << "Enter the third side of the triangle::";
                                cin >> s3;
                                t1.setData(s1 , s2 , s3);                    //initializing data members
                                cout << "The area of the triangle::" << t1.area(s1 , s2 , s3) << endl;
                                break;                                                          //printing the result
```

```
                case 2:
                        cout << endl << "***** Enter the sides of the triangle *****" << endl;
                        cout << "Enter the first side of the triangle::";
                        cin >> s1;
                        cout << "Enter the second side of the triangle::";          //taking the length of sides
                        cin >> s2;
                        t1.setData(s1 , s2);                    //initializing data members
                        cout << "The area of the triangle::" << t1.area(s1 , s2) << endl;
                        break;                                                  //printing the result

                default:
                        cout << "Wrong Input !! Exiting !!";
                        break;
        }

    if(ch==1 || ch==2)
        {
                t1=t2;                                                          //demonstrating assignment
                cout << endl << "********** Checking assignment ***********" << endl;
                if(t1==t2)                                                      //demonstrating equality operator
                        {
                                cout << "Assignment successfull !!";
                        }
                else                                                    //printing if the assignment is successfull
                        {
                                cout << "Not successfull !!";
                        }
        }
    return 0;
}
```

Output:-

Command Prompt

```
C:\Users\harsh\Desktop\AC-1216\Practicales>g++ Practical-10.cpp -o Practical-10.exe

C:\Users\harsh\Desktop\AC-1216\Practicales>Practical-10.exe

****************** MENU ************************
1. Find area using three sides.
2. Find area using two sides.
Enter your choice ::1

***** Enter the sides of the triangle *****
Enter the first side of the triangle::12
Enter the second side of the triangle::12
Enter the third side of the triangle::8
The area of the triangle::45.2548

********** Checking assignment ***********
Assignment successfull !!
C:\Users\harsh\Desktop\AC-1216\Practicales>Practical-10.exe

****************** MENU ************************
1. Find area using three sides.
2. Find area using two sides.
Enter your choice ::2

***** Enter the sides of the triangle *****
Enter the first side of the triangle::12
Enter the second side of the triangle::12
The area of the triangle::72

********** Checking assignment ***********
Assignment successfull !!
C:\Users\harsh\Desktop\AC-1216\Practicales>Practical-10.exe

****************** MENU ************************
1. Find area using three sides.
2. Find area using two sides.
Enter your choice ::6
Wrong Input !! Exiting !!
C:\Users\harsh\Desktop\AC-1216\Practicales>
```

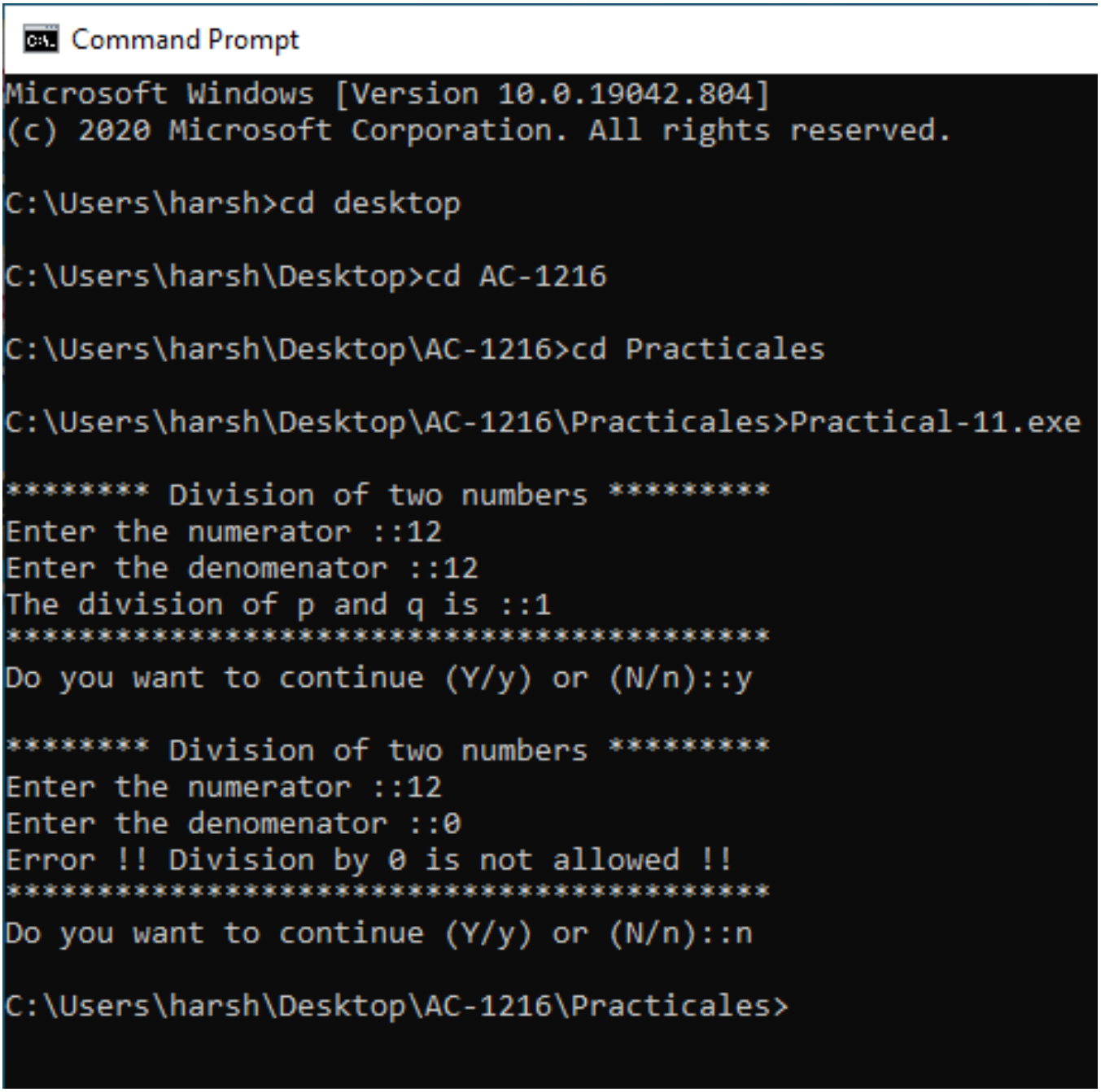
Practical-11

Q. Program to calculate the division of two numbers and perform exception handling .

Code:-

```
 //Harsh Bamotra
//Program to divide to numbers and perform exception handing

#include <iostream>
using namespace std;

int main()
  {
    double p , q;                    //Defining variables
    string ch="Y";
    while(ch=="Y" || ch=="y")
      {
        cout << endl << "******** Division of two numbers *********" << endl;
        cout << "Enter the numerator ::";
        cin >> p;                                          //taking input from the user
        cout << "Enter the denomenator ::";
        cin >> q;                                          //taking input from the user
        try
          {
            if(q==0)                  //checking exception
              {
                throw q;              //throwing the value of q
              }
            else                     //printing division of p and q
              {
                cout << "The division of p and q is ::" << p/q << endl;
              }
          }

        catch(double err)            //catching the thrown value
          {
            cout << "Error !! Division by " << err << " is not allowed !!" << endl;
          }                                                                    //printing the error message

        cout << "*****************************************" << endl;
        cout << "Do you want to continue (Y/y) or (N/n)::";
        cin >> ch;                                                  //asking the user if he wants continue
      }
    return 0;
}
```

Output:-


Command Prompt
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\harsh>cd desktop
C:\Users\harsh\Desktop>cd AC-1216
C:\Users\harsh\Desktop\AC-1216>cd Practicales
C:\Users\harsh\Desktop\AC-1216\Practicales>Practical-11.exe
******** Division of two numbers *********
Enter the numerator ::12
Enter the denomenator ::12
The division of p and q is ::1
****
Do you want to continue (Y/y) or (N/n)::y
******** Division of two numbers *********
Enter the numerator ::12
Enter the denomenator ::0
Error !! Division by 0 is not allowed !!
****
Do you want to continue (Y/y) or (N/n)::n
C:\Users\harsh\Desktop\AC-1216\Practicales>

Practical-12

Q. Program to create a matrix class with matrix functions , overloaded operators and performing exception handling.

Code:-

```
//Harsh Bamotra AC-1216
//Program to perform matrix class with overloading operators and exception handling.

#include <iostream>
using namespace std;

/****************************************** Creating Matrix class ******************************************/
class matrix
  {
        private:                                        //defining  private members
        int row , col;

        public:                                         //defining public members
        int arr[10][10];
        void setData(int n1 , int n2)                   //defining function to initialize the private members
                {
                        row=n1;
                        col=n2;
                }

        void create_arr()                               //defining function to create an array
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << "Enter the elements at index [" << i << "][" << j << "]::";
                                                        cin >> arr[i][j];
                                                }
                                }
                }

        void display_arr()                              //defining function to print the array
                {
                        for(int i=0 ; i<row ; i++)
                                {
                                        cout << endl;
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        cout << arr[i][j] << " ";
                                                }
                                }
                }
```

```
    void trans();                                   //defining function for transpose of the matrix

/************************************** Overloading operators **************************************/

    matrix operator +(matrix m)
        {                                           //overloading + operator for adding two matrix
            matrix temp;                            //defining temp matrix class
            temp.row=m.row;
            temp.col=m.col;
            if(row==m.row && col==m.col)
                {
                for(int i=0 ; i<row ; i++)
                    {
                        for(int j=0 ; j<col ; j++)
                            {
                                temp.arr[i][j]=arr[i][j]+m.arr[i][j];
                            }                                   //adding the elements
                    }                                           //and initializing them in the temp
                }
            else
                {
                cout << "Error !! The order of the matrix are not same. ";
                }

            return temp;                            //returning the result
        }



    matrix operator -(matrix m)
        {                                           //overloading – operator for subtracting two matrix
            matrix temp;                            //defining temp matrix class to store the sum
            temp.row=m.row;
            temp.col=m.col;
            if(row==m.row && col==m.col)
                {
                for(int i=0 ; i<row ; i++)
                    {
                        for(int j=0 ; j<col ; j++)
                            {
                                temp.arr[i][j]=arr[i][j]-m.arr[i][j];
                            }                                   //subtracting the elements
                    }                                           //and initializing them in temp
                }
            else
                {
                cout << "Error !! The order of the matrix are not same. ";
                }

            return temp;                            //returning the result
        }
```

```
        matrix operator *(matrix m)                                     //overloading * operator to multiply two matrix
                {
                        matrix temp;                                    //defining temp matric class
                        temp.row=row;
                        temp.col=m.col;
                        for(int i=0 ; i<row ; i++)
                                {
                                        for(int j=0 ; j<col ; j++)
                                                {
                                                        temp.arr[i][j]=0;    //initializing the elements of temp to 0
                                                }
                                }
                        if(col==m.row)
                                {
                                        for(int i=0 ; i<row ; i++)
                                                {
                                                        for(int j=0 ; j<m.col ; j++)
                                                                {
                                                                        for(int k=0 ; k<col ; k++)
                                                                                {
                                                                                temp.arr[i][j]+=arr[i][k]*m.arr[k][j];
                                                                                }
                                                                }       //multiplying the matrix
                                                }
                                }
                        else
                                {
                                        cout << "Error !! the column of the fisrt matrix not equal to the row of second.";
                                }
                        return temp;                                    //returning the result matrix
                }

    };

void matrix :: trans()                          // defining function trans for transposing matrix
        {
                int arr1[10][10];
                for(int i=0 ; i<row ; i++)
                        {
                                for(int j=0 ; j<col ; j++)
                                        {
                                                arr1[i][j]=arr[j][i];       //transposing the matrix
                                        }
                        }

                cout << "The matrix after transpose::";
                for(int i=0 ; i<row ; i++)
                        {
                                cout << endl;
                                for(int j=0 ; j<col ; j++)
                                        {
                                                cout << arr1[i][j] << " ";
                                        }                                                   //printing the transposed matrix
                        }
        }
```

```
int main(){
	int r1 , r2 , c1 , c2 , x , y;                                  //defining variables
	matrix m1 , m2 , m3 ;                                           //defining matrix class object
	cout << "*****Enter the details of first matrix*****" << endl << endl;
	cout << "Enter the number of row::";
	cin >> r1;                                                      //taking number of rows from the user
	cout << "Enter the number of columns::";
	cin >> c1;                                                      //taking number of columns from the user
	m1.setData(r1 , c1);
	m1.create_arr();                                                //initializing members and creating matrix
	cout << "The matrix you entered::" << endl;
	m1.display_arr();                                               //printing the matrix

	cout << endl << endl << "*****Enter the details of second matrix*****" << endl << endl;
	cout << "Enter the number of row::";
	cin >> r2;                                                      //taking number of rows from the user
	cout << "Enter the number of columns::";
	cin >> c2;                                                      //taking number of columns from the user
	m2.setData(r2 , c2);
	m2.create_arr();                                                //initializing members and creating matrix
	cout << "The matrix you entered::" << endl;
	m2.display_arr();                                               //printing the matrix

	cout << endl << "**********************************************************************" << endl;
	cout << "1.Sum" << endl << "2.Product" << endl << "3.Transpose" << endl << "4.Subracting" << endl ;
	cout << "**********************************************************************" << endl;
	cout << "Enter your choice(1 , 2 , 3 or 4)::";
	cin >> x;                                                       //printing the menu and taking user's choice
	switch (x)                                                      //defining switch case
	{
	case 1:
		try                                                     //handling exception
		{
			if(r1==r2 && c1==c2)
				{
					m3=m1+m2;          //adding and printing the matrix
					cout << "The sum of the matrix::" << endl;
					m3.display_arr();
				}
			else
				{
					throw r1;
				}
		}
		catch(int err)                                          //printing error message
			{
				cout << "Error !! The order of the matrix must be same to perform sum.";
			}
		break;

	case 2:
		try
		{                                               //handling exception
			if(c1==r2)
				{
					m3=m1*m2;                               //multiplying matrix
```

```
                                        cout << "The product of the matrix::" << endl;
                                        m3.display_arr();                                       //printing the result
                                }
                        else
                                {
                                        throw r1;
                                }
                }
                catch(int err)                                                  //printing the error message
                        {
                                cout << "Error !! The column of the first matrix must be same to the row of the second.";
                        }
                break;

        case 3:
                cout << "*****************************************************************" << endl;
                cout << "Which matrix you want to transpose (1 or 2)::";
                cin >> y;
                if(y==1)
                        {
                                m1.trans();
                        }
                else if(y==2)                                                   //transposing and printing matrix
                        {
                                m2.trans();
                        }
                else
                        {
                                cout << "Wrong Input!!" << endl ;
                        }
                break;

        case 4:
                try                                                             //handling exception
                {
                        if(r1==r2 && c1==c2)
                                {
                                        m3=m1-m2;                               subtracting and printing the matrix
                                        cout << "The subtraction of the matrix::";
                                        m3.display_arr();
                                }
                        else
                                {
                                        throw r1;
                                }
                }
                catch(int err)                                  //printing the error message
                        {
                                cout << "Error !! The order of the matrix must be same to perform subtraction.";
                        }
                break;

        default:                                                //exiting message
                cout << "Wrong input !!!";
    }

        return 0 ;
}
```

Output:-

1. Sum

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1
********************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
********************************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
The sum of the matrix::

2 2
2 2
C:\Users\harsh\Desktop>
```

2. Product

Command Prompt

```
Enter the number of columns::3
Enter the elements at index [0][0]::2
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
Enter the elements at index [1][2]::1
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::1
Enter the elements at index [2][2]::1
The matrix you entered::

2 2 1
1 1 1
1 1 1

*****Enter the details of second matrix*****

Enter the number of row::3
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
The matrix you entered::

1 1
1 2
1 2
**********************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**********************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
The product of the matrix::

5 8
3 5
3 5
C:\Users\harsh\Desktop>
```

3. Transpose

Command Prompt

```
Enter the number of row::3
Enter the number of columns::3
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [0][2]::3
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
Enter the elements at index [1][2]::3
Enter the elements at index [2][0]::1
Enter the elements at index [2][1]::2
Enter the elements at index [2][2]::3
The matrix you entered::

1 2 3
1 2 3
1 2 3

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2
********************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
********************************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
********************************************************************************
Which matrix you want to transpose (1 or 2)::1
The matrix after transpose::
1 1 1
2 2 2
3 3 3
C:\Users\harsh\Desktop>
```

Command Prompt

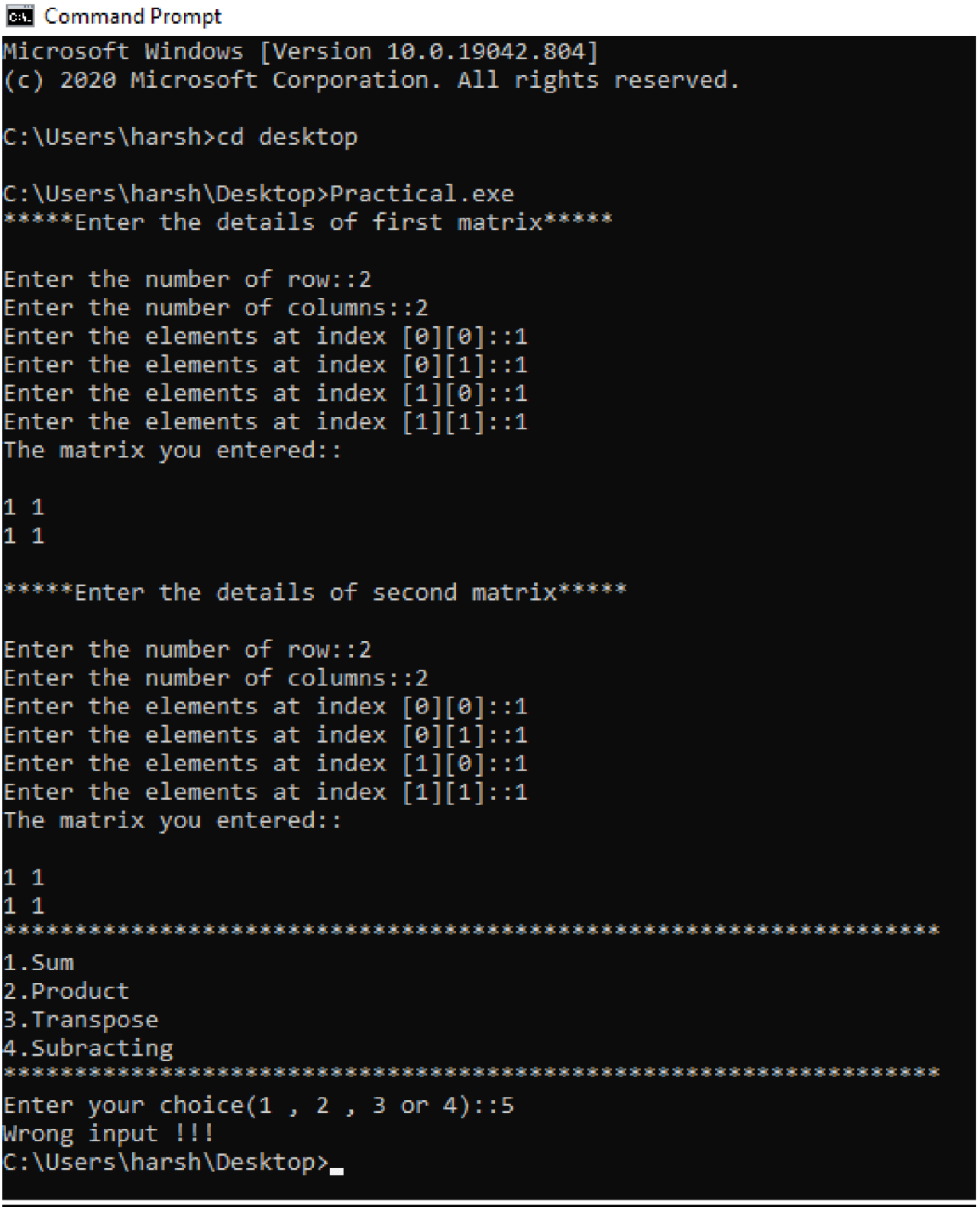
```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::2
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::2
The matrix you entered::

1 2
1 2
********************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
********************************************************************************
Enter your choice(1 , 2 , 3 or 4)::3
********************************************************************************
Which matrix you want to transpose (1 or 2)::2
The matrix after transpose::
1 1
2 2
C:\Users\harsh\Desktop>
```

4. Subtraction

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::4
The subtraction of the matrix::
0 0
0 0
C:\Users\harsh\Desktop>_
```

5. Exception handling



Exception in main menu

Command Prompt

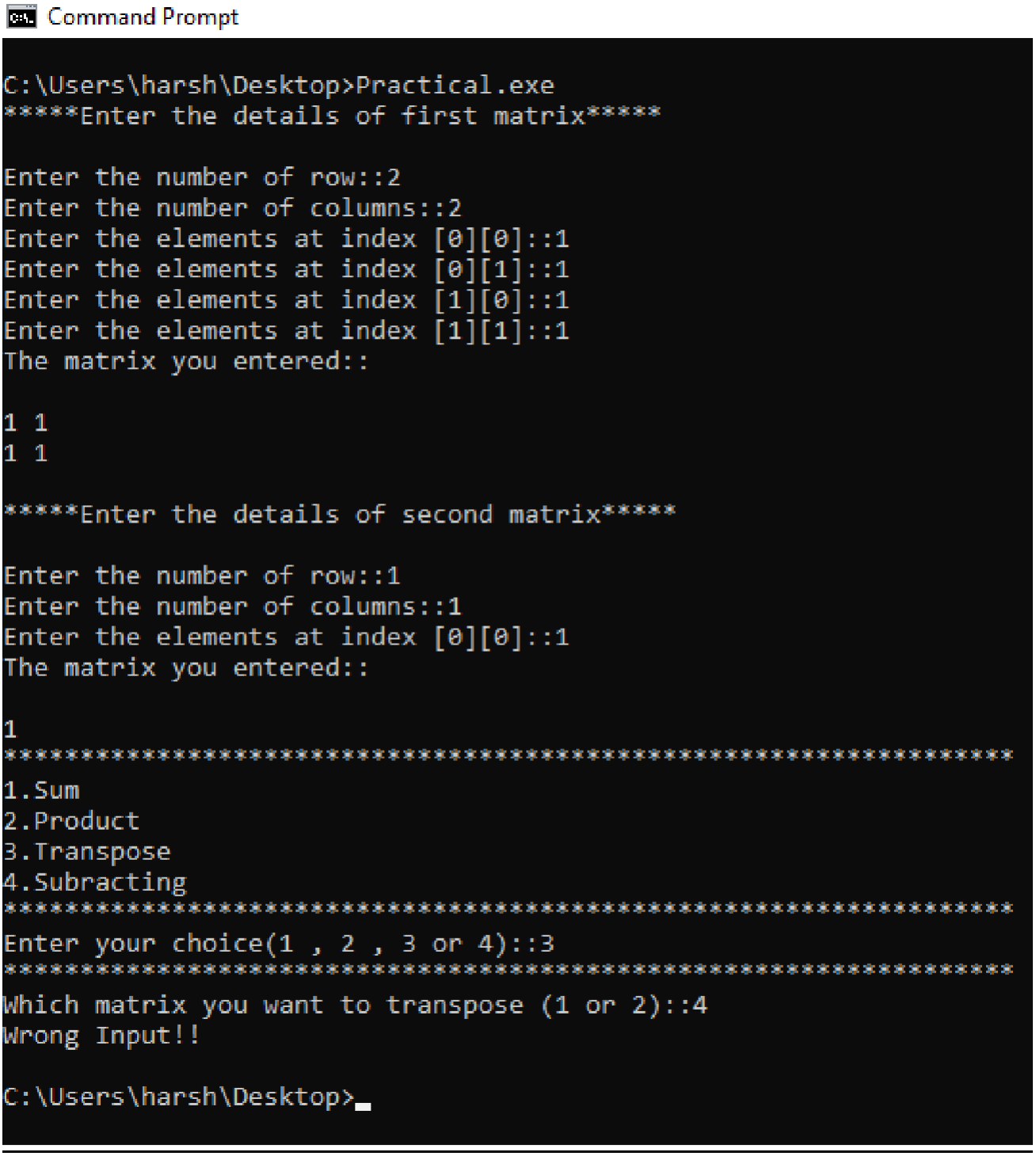
```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::1
Enter the elements at index [0][0]::1
The matrix you entered::

1
********************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
********************************************************************************
Enter your choice(1 , 2 , 3 or 4)::1
Error !! The order of the matrix must be same to perform sum.
C:\Users\harsh\Desktop>
```

Exception in case of sum

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::2
Error !! The column of the first matrix must be same to the row of the second.
C:\Users\harsh\Desktop>
```

Exception in case of product



Exception in case of transpose

Command Prompt

```
C:\Users\harsh\Desktop>Practical.exe
*****Enter the details of first matrix*****

Enter the number of row::2
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
Enter the elements at index [1][0]::1
Enter the elements at index [1][1]::1
The matrix you entered::

1 1
1 1

*****Enter the details of second matrix*****

Enter the number of row::1
Enter the number of columns::2
Enter the elements at index [0][0]::1
Enter the elements at index [0][1]::1
The matrix you entered::

1 1
**************************************************************************
1.Sum
2.Product
3.Transpose
4.Subracting
**************************************************************************
Enter your choice(1 , 2 , 3 or 4)::4
Error !! The order of the matrix must be same to perform subtraction.
C:\Users\harsh\Desktop>
```

Exception in case of subtraction

Practical-13

Q. Program to create a student class and take details of five students and save them in a text file .

Code:-

```
//Harsh Bamotra
//Program to write the details of student in a text file

#include <iostream>
#include <fstream>
using namespace std;
class student{                                          //Defining class student
      private:                                         //defining private members
      string roll_no , name , Class;
      int year , t_marks;

      public:                                 //defining public members
      void getdata()                         //defining function to input data from user
        {
          cout << "Enter your name ::";
          getline(cin , name);
          fflush;
          cout << "Enter your roll no ::";
          cin >> roll_no;
          cout << "Enter your class ::";
          cin >> Class;
          cout << "Enter the year ::";
          cin >> year;
          cout << "Enter your Total marks ::";
          cin >> t_marks;
          cin.ignore();

        }

      void write()                                          //defining function to write in the text file
        {                                                  //defining output stream cursor to write in the text file
          ofstream out("sample.txt" , ios::app);
          out << "***** Details of the student *****\n";
          out << "Name ::" << name << "\n" ;
          out << "Roll No. ::" << roll_no << "\n" ;
          out << "Class ::" << Class << "\n" ;
          out << "Year ::" << year << "\n" ;
          out << "Total marks ::" << t_marks << "\n";
          out << "**********************************\n\n";
        }
```

```
    void display()                                  //defining function to display inputted data
      {
        cout << "Roll No. ::" << roll_no << endl;
        cout << "Name ::" << name << endl;
        cout << "Class ::" << Class << endl;
        cout << "Year ::" << year << endl;
        cout << "Total marks ::" << t_marks;
      }

  };

int main()
  {
      student arr[5];                                   //defining class student object

      for(int i=0 ; i<5 ; i++)                          //loop to enter data of 5 students
          {
              cout << "***** Enter the details of the student " << i+1 << " *****" << endl;
              arr[i].getdata();
              cout << endl;                                 //taking input from the user
          }

       for(int i=0 ; i<5 ; i++)                         //loop to print and write data of 5 students
          {
              cout << "***** Details of the student " << i+1 << " *****" << endl;
              arr[i].display();                          //printing the data to the user
              arr[i].write();                           //writing data in the text file
              cout << endl;
              cout << "**** Saved this data successfully ****" << endl << endl;
          }
      return 0;

  }
```

Output:-

1. Giving input to the program

Command Prompt

```
C:\Users\harsh\Desktop>Practical-13.exe
***** Enter the details of the student 1 *****
Enter your name ::Sam
Enter your roll no ::1
Enter your class ::12
Enter the year ::2021
Enter your Total marks ::87

***** Enter the details of the student 2 *****
Enter your name ::Ram lal
Enter your roll no ::2
Enter your class ::12
Enter the year ::2021
Enter your Total marks ::89

***** Enter the details of the student 3 *****
Enter your name ::Sham lal
Enter your roll no ::3
Enter your class ::12
Enter the year ::2021
Enter your Total marks ::76

***** Enter the details of the student 4 *****
Enter your name ::Harsh Bamotra
Enter your roll no ::4
Enter your class ::12
Enter the year ::2021
Enter your Total marks ::100

***** Enter the details of the student 5 *****
Enter your name ::Ipshita Mahajan
Enter your roll no ::5
Enter your class ::12
Enter the year ::2021
Enter your Total marks ::100
```

2. Showing the data to the user

```
***** Details of the student 1 *****
Roll No. ::1
Name ::Sam
Class ::12
Year ::2021
Total marks ::87
**** Saved this data successfully ****
```

```
***** Details of the student 2 *****
Roll No. ::2
Name ::Ram lal
Class ::12
Year ::2021
Total marks ::89
**** Saved this data successfully ****

***** Details of the student 3 *****
Roll No. ::3
Name ::Sham lal
Class ::12
Year ::2021
Total marks ::76
**** Saved this data successfully ****

***** Details of the student 4 *****
Roll No. ::4
Name ::Harsh Bamotra
Class ::12
Year ::2021
Total marks ::100
**** Saved this data successfully ****

***** Details of the student 5 *****
Roll No. ::5
Name ::Ipshita Mahajan
Class ::12
Year ::2021
Total marks ::100
**** Saved this data successfully ****
```

3. Data saved in the text file

sample - Notepad

File Edit Format View Help

```
***** Details of the student *****
Name ::Sam
Roll No. ::1
Class ::12
Year ::2021
Total marks ::87
**********************************

***** Details of the student *****
Name ::Ram lal
Roll No. ::2
Class ::12
Year ::2021
Total marks ::89
**********************************

***** Details of the student *****
Name ::Sham lal
Roll No. ::3
Class ::12
Year ::2021
Total marks ::76
**********************************

***** Details of the student *****
Name ::Harsh Bamotra
Roll No. ::4
Class ::12
Year ::2021
Total marks ::100
**********************************

***** Details of the student *****
Name ::Ipshita Mahajan
Roll No. ::5
Class ::12
Year ::2021
Total marks ::100
**********************************
```

Practical-14

Q. Program to copy the contents of a file to another after removing all the whitespaces.

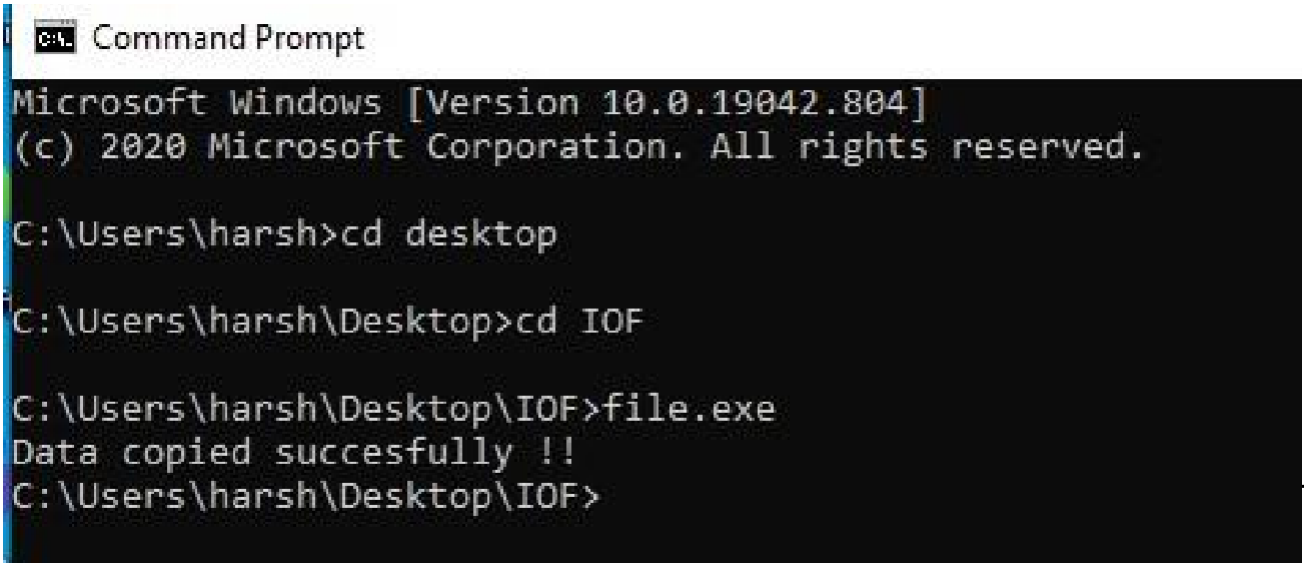
Code:-

```
 //Harsh Bamotra
//Program to copy the contents of a file to other after removing spaces.

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

int main()
  {
        string str;                     //defining string str
        ifstream in("a.txt");                       //defining cursor for reading from the text file
        if(in)
                {
                        cout << "Data copied succesfully !!";
                }
        else
                {
                        cout << "File not found !!";
                }

        ofstream out("a1.txt");                 //defining cursor for writing in the text file
        while(in)
                {
                in >> str;                                  //reading from the file a
                if(str!=" ")                                //checking for whitespaces
                        {
                                        out << str;             //writing in the file
                        }

                if(in.eof())                                //checking the end of the file
                        {
                                        break;          //breaking the loop at the end of the file
                        }
                }

        return 0;
  }
```

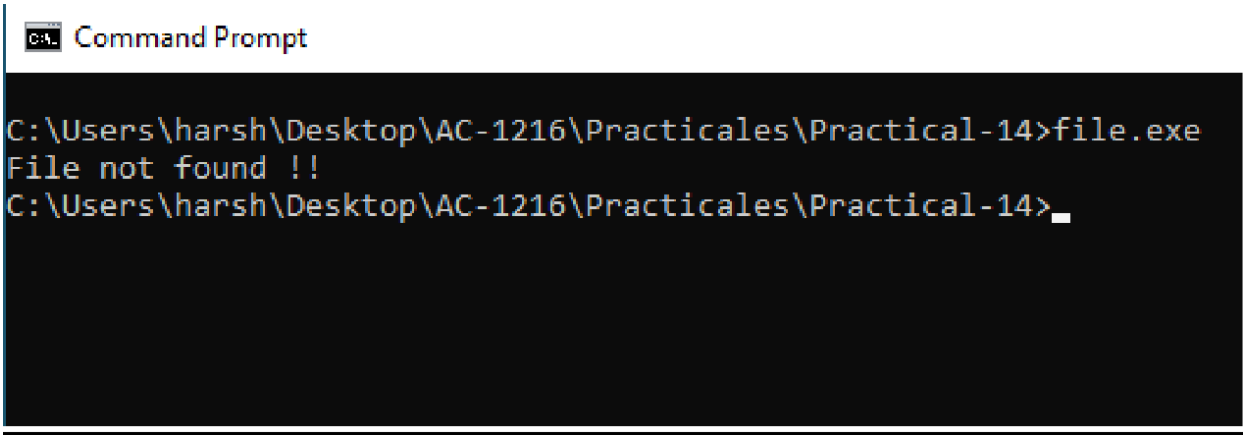
Output:-

1. **Message in the console .**

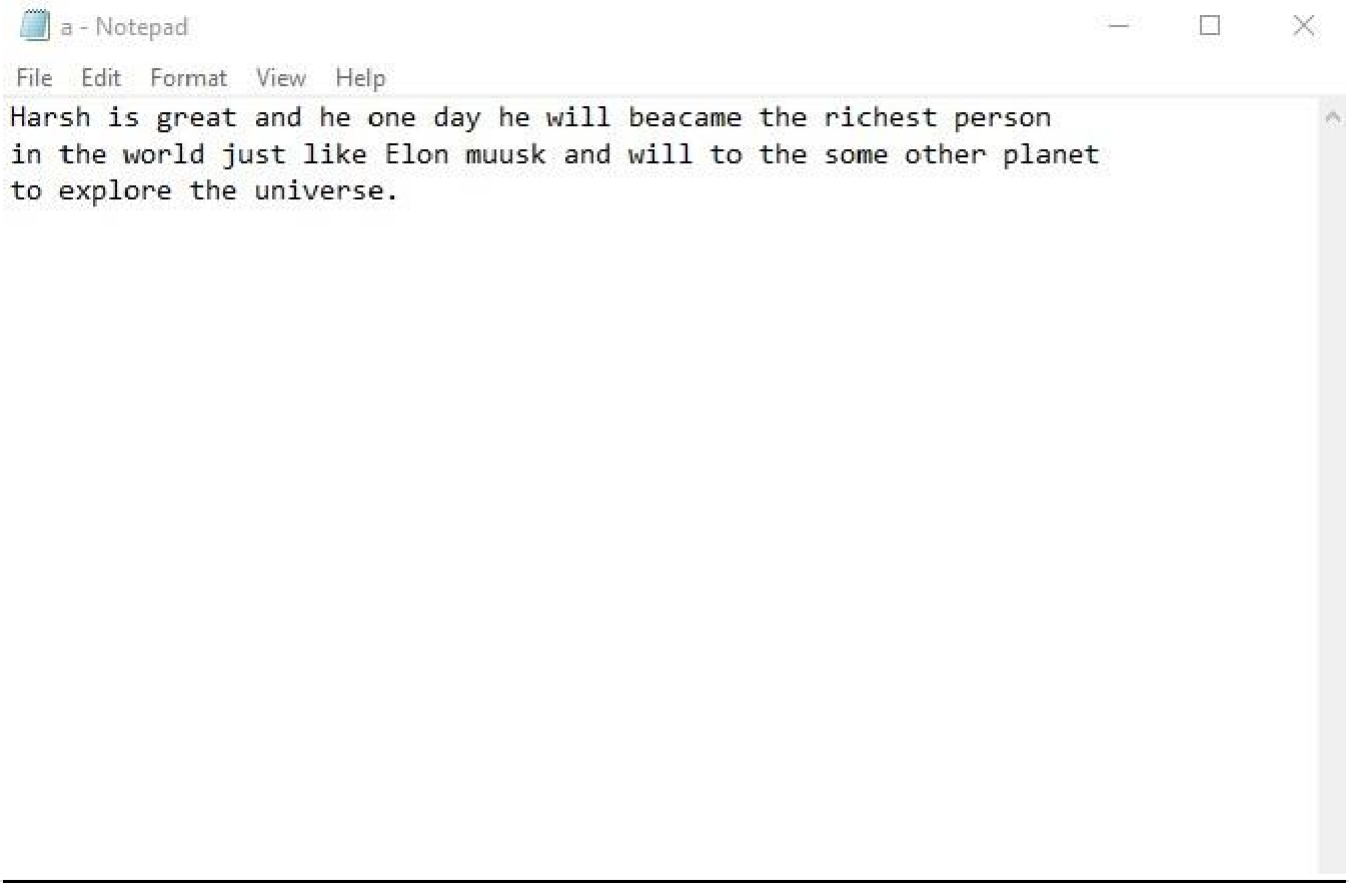
   When file is present in the directory



When file is present not in the directory



2. Both files before coping



3. File after copying the content