Atma Ram Sanatan Dharma College
University of Delhi

Programming Fundamentals Using C++

Practical File

Submitted By

KHUSHAL SACHDEVA
20/88044
BSc. (Hons.) Computer Science (SEM - 1)

Submitted To

Ms Parul Jain
Department of Computer Science

INDEX

16	Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain respective information in the classes and create, display and delete objects of these two classes. (Use Run Time Polymorphism)
17	Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator
18	Copy the contents of one text file to another file, after removing all whitespaces
19	Create a class Box containing length, breadth and height. Write a program which takes input from user for length, breadth and height to test the class
20	Create a class Length containing feet and inch. Write a program which takes input from user for feet and inch to test the class

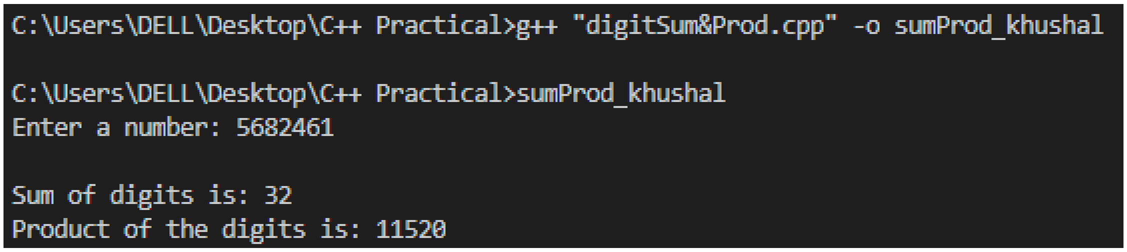
Practical - 1

Objective : Write a program to print the sum and product of digits of an integer.

Code :

```
/*
QUESTION-1:  Write a program to print the sum and product of digits of an integer.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
    int n, sum = 0, prod = 1;
    cout << "Enter a number: ";
    cin >> n;
    while (n > 0)
    {
        int rem = n % 10;
        prod = prod * rem;
        sum += rem;
        n = n / 10;
    }
cout <<"\nSum of digits is: "<< sum <<"\nProduct of the digits is: "<<
prod;
return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ "digitSum&Prod.cpp" -o sumProd_khushal

C:\Users\DELL\Desktop\C++ Practical>sumProd_khushal
Enter a number: 5682461

Sum of digits is: 32
Product of the digits is: 11520
```

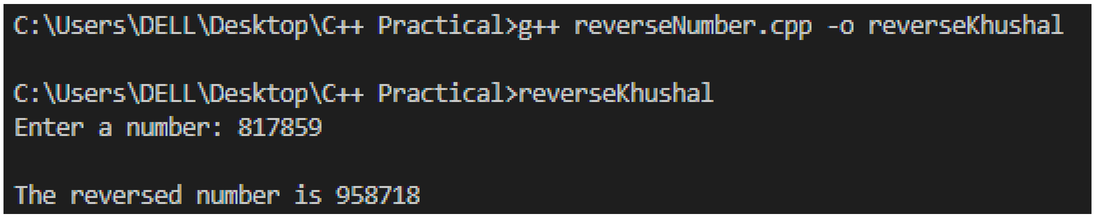
Practical - 2

Objective : Write a program to reverse a number.

Code :

```
/*
QUESTION-2: Write a program to reverse a number.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
    int n, rev = 0;
    cout << "Enter a number: ";
    cin >> n;
    while (n > 0)
    {
        int rem = n % 10;
        rev = rev * 10 + rem;
        n = n / 10;
    }
    cout << "\nThe reversed number is " << rev;
return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ reverseNumber.cpp -o reverseKhushal

C:\Users\DELL\Desktop\C++ Practical>reverseKhushal
Enter a number: 817859

The reversed number is 958718
```

Practical - 3

Objective : Write a program to compute the sum up to the n terms of the following series:

$$S=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}+\frac{1}{9}+\cdots$$

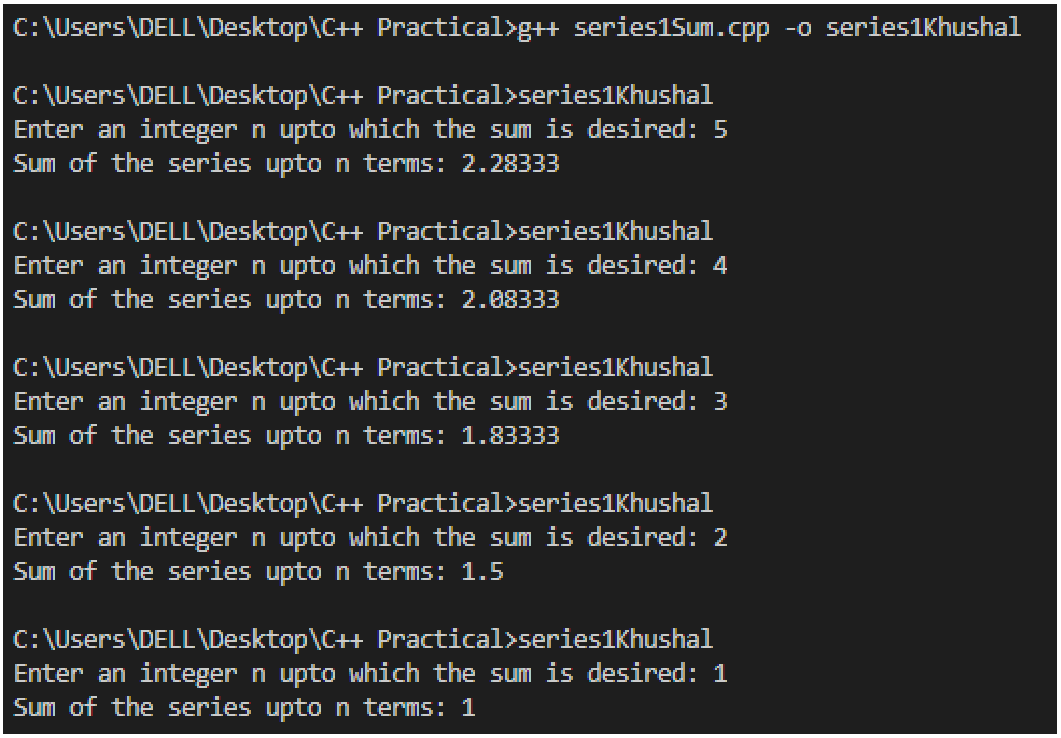
Code :

```
/*
QUESTION-3: Write a program to compute the sum up to the n terms of
the following series:
```

$$S=1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}+\frac{1}{9}+\cdots$$

```
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
    int n;
    float sum = 0;
    cout <<"Enter an integer n upto which the sum is desired: ";
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        sum += ((float)1 / (float)i);
    }
    cout << "Sum of the series upto n terms: " << sum << endl;
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ series1Sum.cpp -o series1Khushal

C:\Users\DELL\Desktop\C++ Practical>series1Khushal
Enter an integer n upto which the sum is desired: 5
Sum of the series upto n terms: 2.28333

C:\Users\DELL\Desktop\C++ Practical>series1Khushal
Enter an integer n upto which the sum is desired: 4
Sum of the series upto n terms: 2.08333

C:\Users\DELL\Desktop\C++ Practical>series1Khushal
Enter an integer n upto which the sum is desired: 3
Sum of the series upto n terms: 1.83333

C:\Users\DELL\Desktop\C++ Practical>series1Khushal
Enter an integer n upto which the sum is desired: 2
Sum of the series upto n terms: 1.5

C:\Users\DELL\Desktop\C++ Practical>series1Khushal
Enter an integer n upto which the sum is desired: 1
Sum of the series upto n terms: 1
```

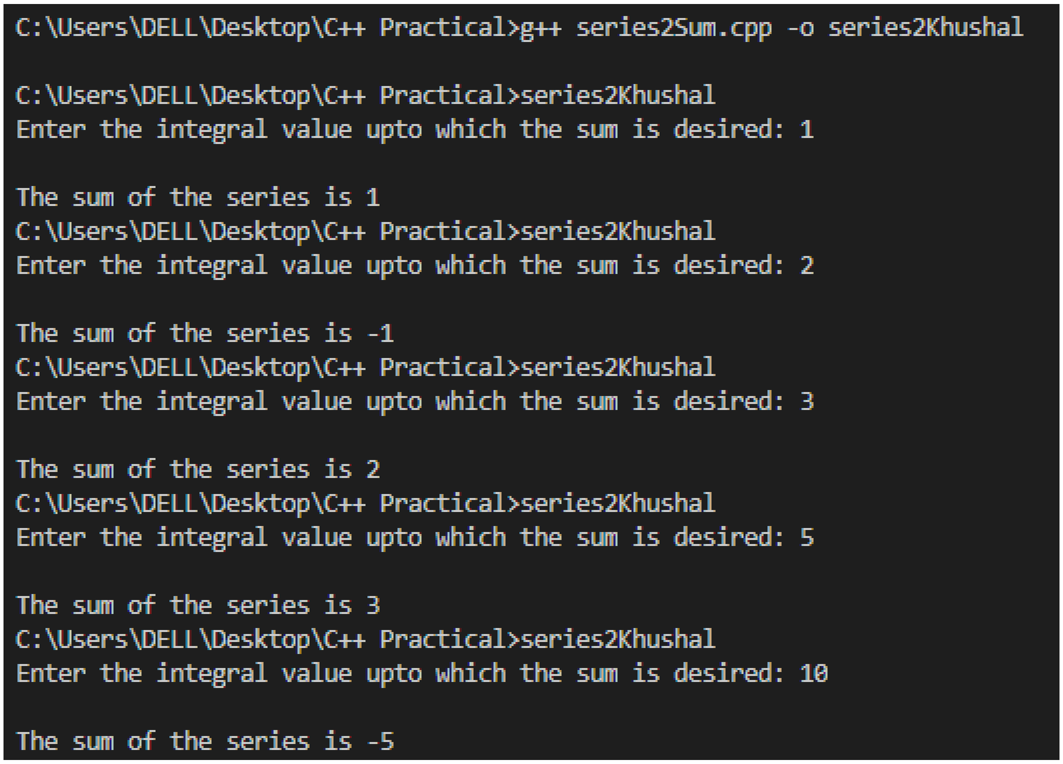
Practical - 4

Objective : Write a program to compute the sum up to the n terms of the following series: S=1 – 2 + 3 – 4 +5−6+7−8+9−⋯

Code :

```
/*
QUESTION-4:  Write a program to compute the sum up to the n terms of the following series:
                                        S=1 – 2 + 3 – 4 +5−6+7−8+9−⋯
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
   int n, sum = 0;
   cout << "Enter the integral value upto which the sum is
   desired: ";
    cin >> n;
    while (n > 0)
    {
        if (n % 2 == 0)
            sum -= n;
        else
            sum += n;
        n--;
    }
    cout << "\nThe sum of the series is " << sum;
   return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ series2Sum.cpp -o series2Khushal

C:\Users\DELL\Desktop\C++ Practical>series2Khushal
Enter the integral value upto which the sum is desired: 1

The sum of the series is 1
C:\Users\DELL\Desktop\C++ Practical>series2Khushal
Enter the integral value upto which the sum is desired: 2

The sum of the series is -1
C:\Users\DELL\Desktop\C++ Practical>series2Khushal
Enter the integral value upto which the sum is desired: 3

The sum of the series is 2
C:\Users\DELL\Desktop\C++ Practical>series2Khushal
Enter the integral value upto which the sum is desired: 5

The sum of the series is 3
C:\Users\DELL\Desktop\C++ Practical>series2Khushal
Enter the integral value upto which the sum is desired: 10

The sum of the series is -5
```

Practical - 5

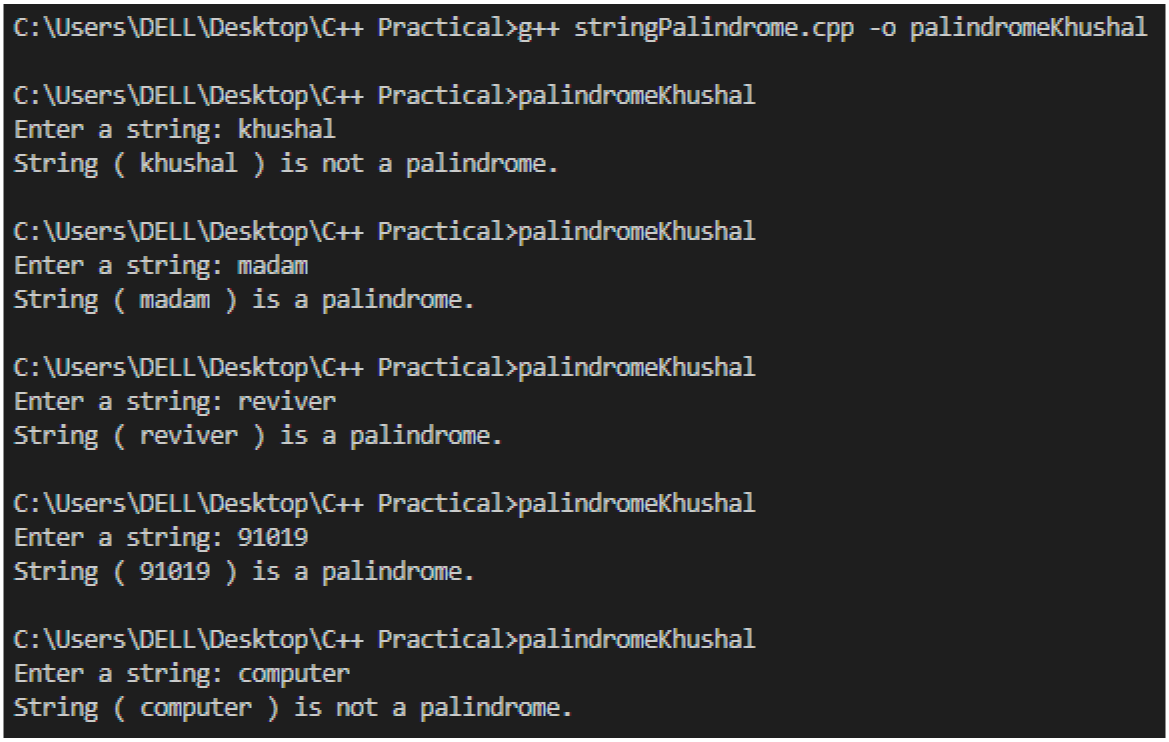
Objective : Write a function that checks whether a given string is a Palindrome or not and use it to check whether the string entered by the user is a Palindrome.

Code :

```
/*
QUESTION-5: Write a function that checks whether a given string is a
Palindrome or not and use it to check whether the string entered by the
user is a Palindrome.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;

void checkPalindrome(char s[])
{
    int i = 0;
    while (s[i] != '\0')
    {
        i++;
    }
    i--;
    for (int j = 0, k = i; j <= i / 2; j++, k--)
    {
        if (s[j] != s[k])
        {
          cout << "String ( " << s << " ) is not a palindrome."<<endl;
          return;
        }
    }
    cout << "String ( " << s << " ) is a palindrome."<<endl;
    return;
```

```
int main()
{
    char str[255];
    cout << "Enter a string: ";
    cin >> str;
    checkPalindrome(str);
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ stringPalindrome.cpp -o palindromeKhushal

C:\Users\DELL\Desktop\C++ Practical>palindromeKhushal
Enter a string: khushal
String ( khushal ) is not a palindrome.

C:\Users\DELL\Desktop\C++ Practical>palindromeKhushal
Enter a string: madam
String ( madam ) is a palindrome.

C:\Users\DELL\Desktop\C++ Practical>palindromeKhushal
Enter a string: reviver
String ( reviver ) is a palindrome.

C:\Users\DELL\Desktop\C++ Practical>palindromeKhushal
Enter a string: 91019
String ( 91019 ) is a palindrome.

C:\Users\DELL\Desktop\C++ Practical>palindromeKhushal
Enter a string: computer
String ( computer ) is not a palindrome.
```

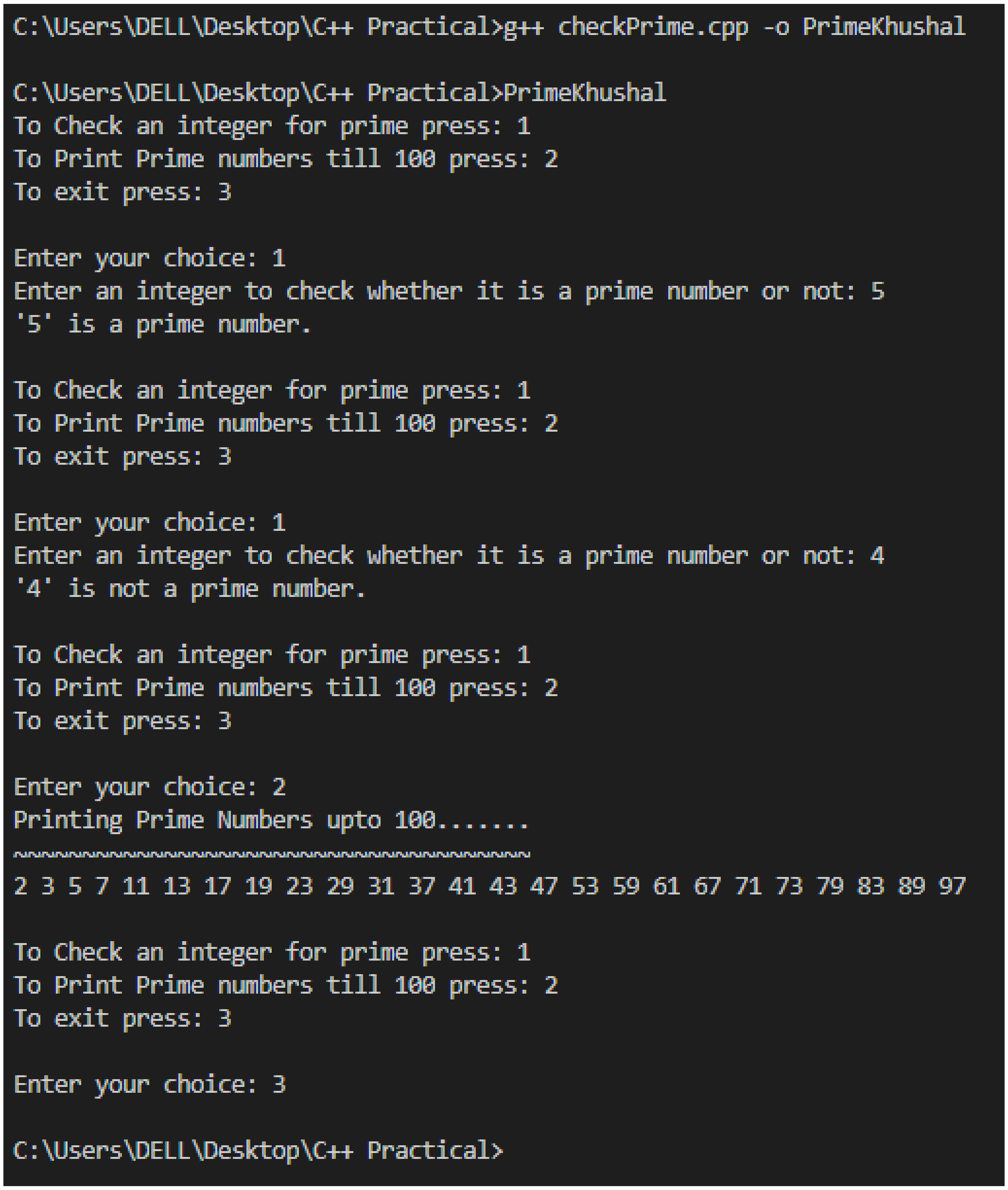
Practical - 6

Objective : Write a function to find whether a given no. is Prime or not. Use the same to generate prime numbers less than 100.

Code :

```
/*
QUESTION-6: Write a function to find whether a given no. is Prime or not. Use
the same to generate prime numbers less than 100.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;

bool checkPrime(int num)
{
    if (num <= 1)
        return false;
    else
      for (int i = 2; i < num; i++)
       {
        if (num % i == 0) //checking by dividing the number by 2 till number
                          // itself
           return false;
       }
    return true;
}

int main()
{
    int x;
    do
    {
        cout << "To Check an integer for prime press: 1" << endl
             << "To Print Prime numbers till 100 press: 2" << endl
             << "To exit press: 3" << endl
             << endl
             << "Enter your choice: ";
        cin >> x;
        switch (x)
        {
        case (1):
        {
          int n;
          cout << "Enter an integer to check whether it is a prime number or
          not: ";
            cin >> n;

            if (checkPrime(n))
                cout << "'" << n << "'"
                     << " is a prime number." << endl
                     << endl;
            else
                cout << "'" << n << "'"
                     << " is not a prime number." << endl
                     << endl;
            break;
        }
        case (2):
        {
            cout << "Printing Prime Numbers upto 100......." << endl;
            cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
            int i = 2;
            while (i < 100)
            {
                if (checkPrime(i))
                    cout << i << " ";
                i++;
            }
            cout << endl
                 << endl;
            break;
        }
        }
    } while (x < 3);
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ checkPrime.cpp -o PrimeKhushal

C:\Users\DELL\Desktop\C++ Practical>PrimeKhushal
To Check an integer for prime press: 1
To Print Prime numbers till 100 press: 2
To exit press: 3

Enter your choice: 1
Enter an integer to check whether it is a prime number or not: 5
'5' is a prime number.

To Check an integer for prime press: 1
To Print Prime numbers till 100 press: 2
To exit press: 3

Enter your choice: 1
Enter an integer to check whether it is a prime number or not: 4
'4' is not a prime number.

To Check an integer for prime press: 1
To Print Prime numbers till 100 press: 2
To exit press: 3

Enter your choice: 2
Printing Prime Numbers upto 100.......
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

To Check an integer for prime press: 1
To Print Prime numbers till 100 press: 2
To exit press: 3

Enter your choice: 3

C:\Users\DELL\Desktop\C++ Practical>
```

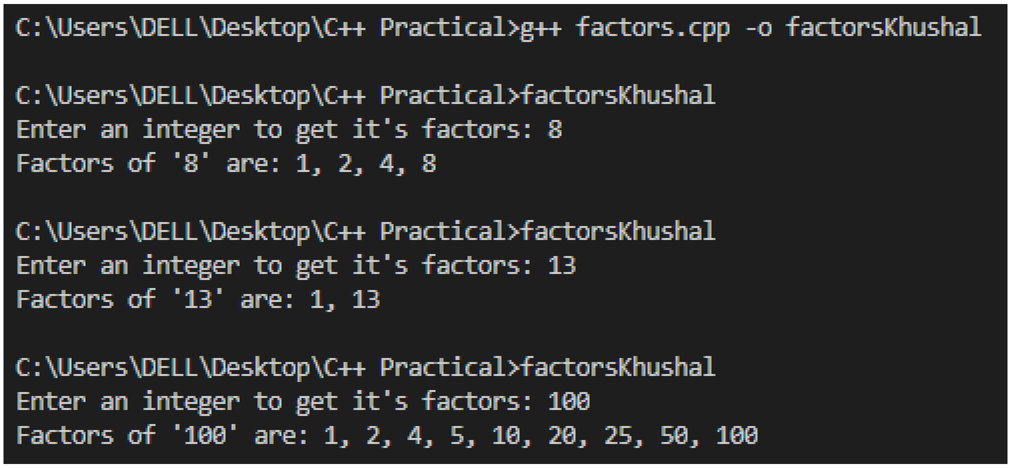
Practical - 7

Objective : Write a program to compute the factors of a given number.

Code :

```
/*
QUESTION-7: Write a program to compute the factors of a given number.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
    int num, factor;
    cout << "Enter an integer to get it's factors: ";
    cin >> num;
    cout << "Factors of ‘" << num << "’ are: ";
    for (int i = 1; i <= num; i++)
    {
        if (num % i == 0)
        {
            cout << i;
            if (i < num)
            {
                cout << ", ";
            }
        }
     }
    cout << endl;
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ factors.cpp -o factorsKhushal

C:\Users\DELL\Desktop\C++ Practical>factorsKhushal
Enter an integer to get it's factors: 8
Factors of '8' are: 1, 2, 4, 8

C:\Users\DELL\Desktop\C++ Practical>factorsKhushal
Enter an integer to get it's factors: 13
Factors of '13' are: 1, 13

C:\Users\DELL\Desktop\C++ Practical>factorsKhushal
Enter an integer to get it's factors: 100
Factors of '100' are: 1, 2, 4, 5, 10, 20, 25, 50, 100
```

Practical - 8

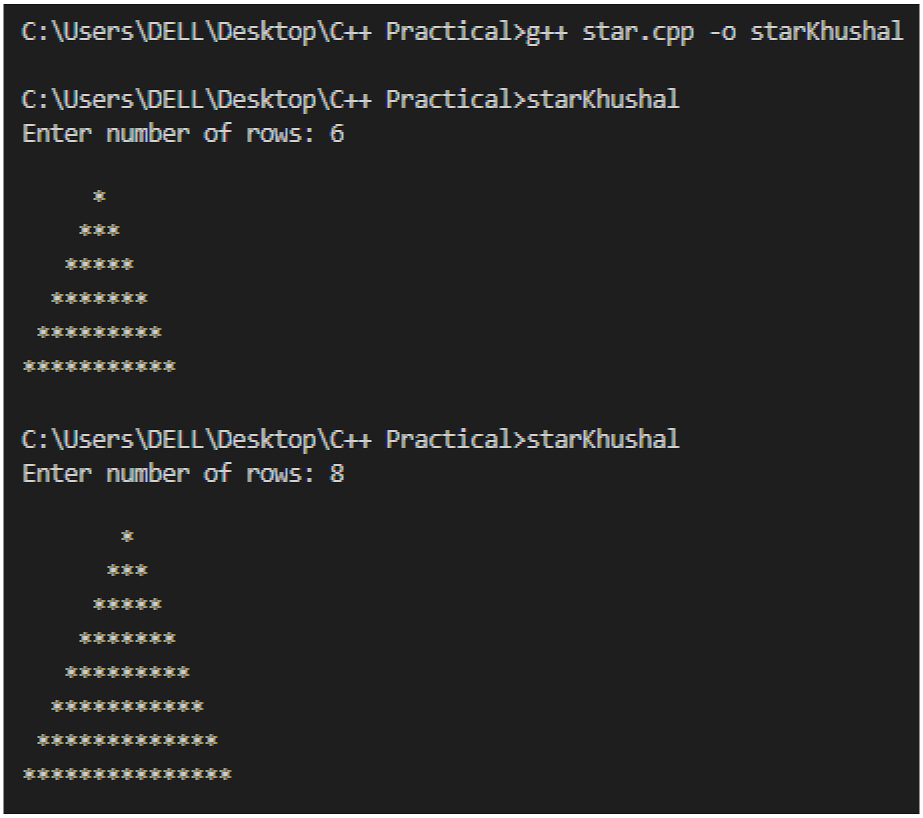
Objective : Write a program to print the triangle of stars as follows (take number of lines from user):

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

Code :

```
/*
QUESTION-8: Write a program to print the triangle of stars as follows
            (take number of lines from user):
                                    *
                                   ***
                                  *****
                                 *******
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
  int n;
  cout << "Enter number of rows: ";
  cin >> n;
  cout << endl;
  for (int i = 1; i <= n; i++) //loop for each line
  {
    for (int j = 1; j <= n - i; j++) //number of spaces = n-i
    {
      cout << ' ';
    }
    for (int k = 1; k <= 2 * i - 1; k++) //number of stars = 2i-1(odd nos.)
        {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ star.cpp -o starKhushal

C:\Users\DELL\Desktop\C++ Practical>starKhushal
Enter number of rows: 6

     *
    ***
   *****
  *******
 *********
***********

C:\Users\DELL\Desktop\C++ Practical>starKhushal
Enter number of rows: 8

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

Practical - 9

Objective : Write a menu driven program to perform following actions on an array entered by the user:
(a) Print the even-valued elements.
(b) Print the odd-valued elements
(c) Calculate and print the sum and average of the elements of the array.
(d) Print the maximum and minimum elements of the array.
(e) Remove duplicates from the array.
(f) Print the array in reverse order.
The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter the array and to quit the program.

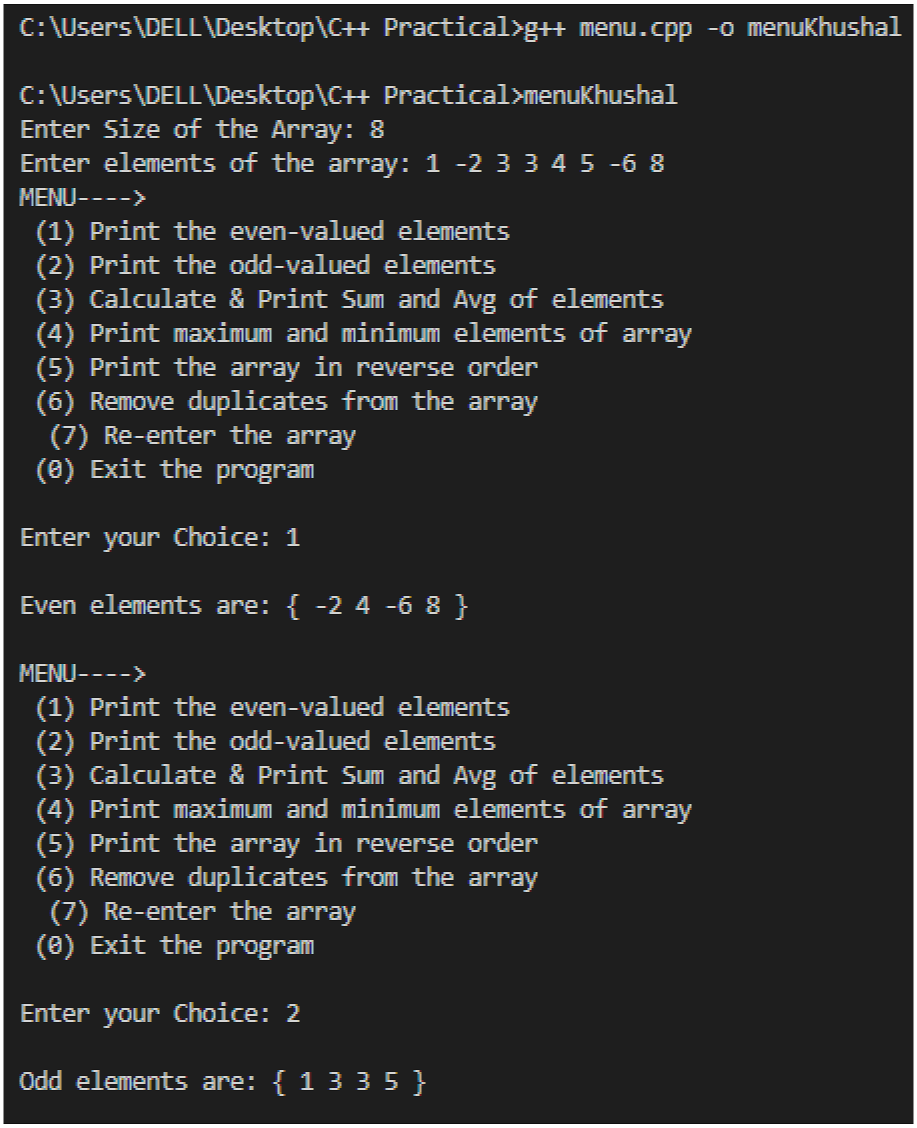
Code :

```
/**
 * Write a menu driven program to perform following actions on
 * an array entered by the user:
 * (a) Print the even-valued elements.
 * (b) Print the odd-valued elements.
 * (c) Calculate and print the sum and average of the elements of the array.
 * (d) Print the maximum and minimum elements of the array.
 * (e) Remove the duplicates from the array.
 * (f) Print the array in reverse order.
 * The program should present a menu to the user and ask for one of the options.
 * The menu should also include options to re-enter and to quit the program.
 * Written by: Khushal Sachdeva
*/
#include <iostream>
using namespace std;

int main()
{
    int choice, size;
    int array[30] = {0};
    cout << "Enter Size of the Array: ";
    cin >> size;
    cout << "Enter elements of the array: ";
    for (int i = 0; i < size; i++)
    {
        cin >> array[i];
    }
    do
    {
        int min = array[0];
        int max = array[0];
        cout << "MENU---->" << endl;
        cout << " (1) Print the even-valued elements\n"
             << " (2) Print the odd-valued elements\n"
             << " (3) Calculate & Print Sum and Avg of elements\n"
             << " (4) Print maximum and minimum elements of array\n"
             << " (5) Print the array in reverse order\n"
             << " (6) Remove duplicates from the array\n "
             << " (7) Re-enter the array\n"
             << " (0) Exit the program\n";
        cout << "\nEnter your Choice: ";
        cin >> choice;
        switch (choice)
        {
        case 1:
        {
            cout << "\nEven elements are: { ";
            for (int j = 0; j < size; j++)
            {
                if ((array[j] % 2) == 0)
                {
                    cout << array[j] << ' ';
                }
            }
            cout << "}" << endl
                 << endl;
            break;
        }
        case 2:
        {
            cout << "\nOdd elements are: { ";
            for (int j = 0; j < size; j++)
            {
                if ((array[j] % 2) != 0)
                {
                    cout << array[j] << ' ';
                }
            }
            cout << "}" << endl
                 << endl;
            break;
        }
        case 3:
        {
            int sum = 0, avg = 0;
            for (int i = 0; i < size; i++)
            {
                sum += array[i];
            }
            cout << "\nSum of elements: ";
            cout << sum;
            avg = (float)sum / size;
            cout << "\nAverage of elements: ";
            cout << avg << endl
                 << endl;
            break;
        }
        case 4:
        {

            for (int i = 0; i < size; i++)
            {
                if (array[i] > max)
```

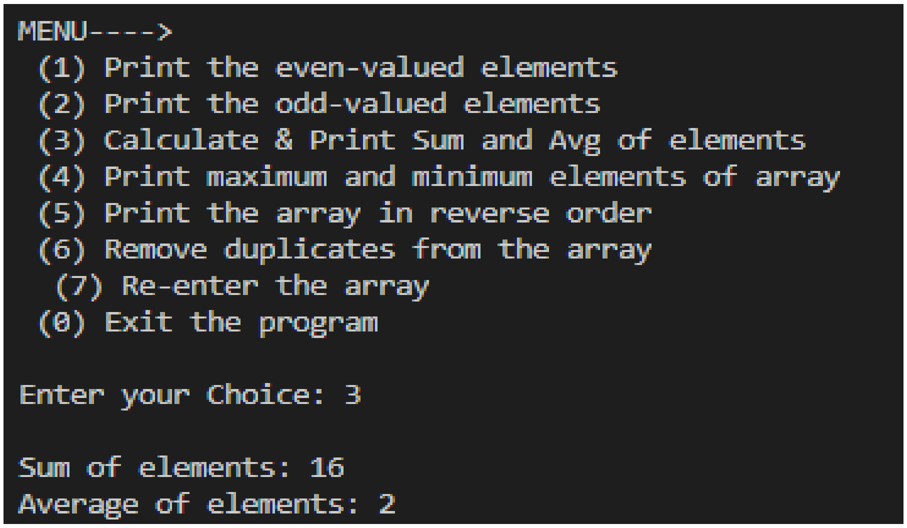
```
            max = array[i];
        if (array[i] < min)
            min = array[i];
    }
    cout << "\nMinimum= " << min;
    cout << "\nMaximum= " << max << endl
         << endl;
    break;
}
case 5:
{
    int i;
    cout << "\nReversed Array: [ ";
    for (int i = size - 1; i >= 0; i--)
        cout << array[i] << ' ';
    cout << "]\n\n";
    break;
}
case 6:
{
    int i, j, k;
    int count = 0;
    for (i = 0; i < size; ++i)
        for (j = i + 1; j < size;)
        {
            if (array[i] == array[j])
            {
                count++;
                for (k = j; k < size - 1; ++k)
                    array[k] = array[k + 1];

                --size;
            }
            else
                ++j;
        }
    cout << "\nDuplicates Found: " << count;
```

```
            cout << "\nRemoved all duplicates ...";
            cout << "\nNew Array: [";
            for (int i = 0; i < size; i++)
                cout << array[i] << ' ';
            cout << "]\n\n";
            break;
        }
        case 7:
        {
            cout << endl;
            cout << "Enter Size of the Array: ";
            cin >> size;
            cout << "Enter elements of the array: ";
            for (int i = 0; i < size; i++)
            {
                cin >> array[i];
            }
        }
        case 0:
            break;
        default:
        {
            cout << "\nInvalid Option ";
            break;
        }
        }
    } while (choice != 0);

    return 0;
}
```

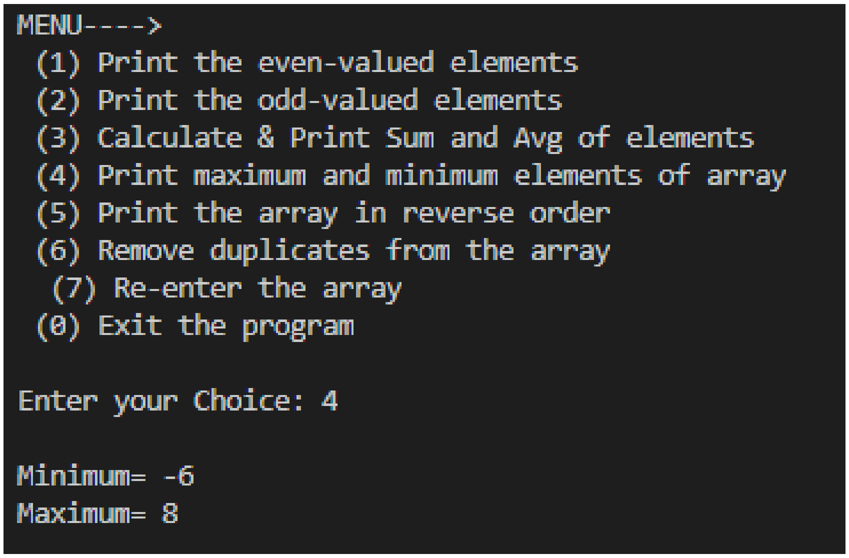
Output :



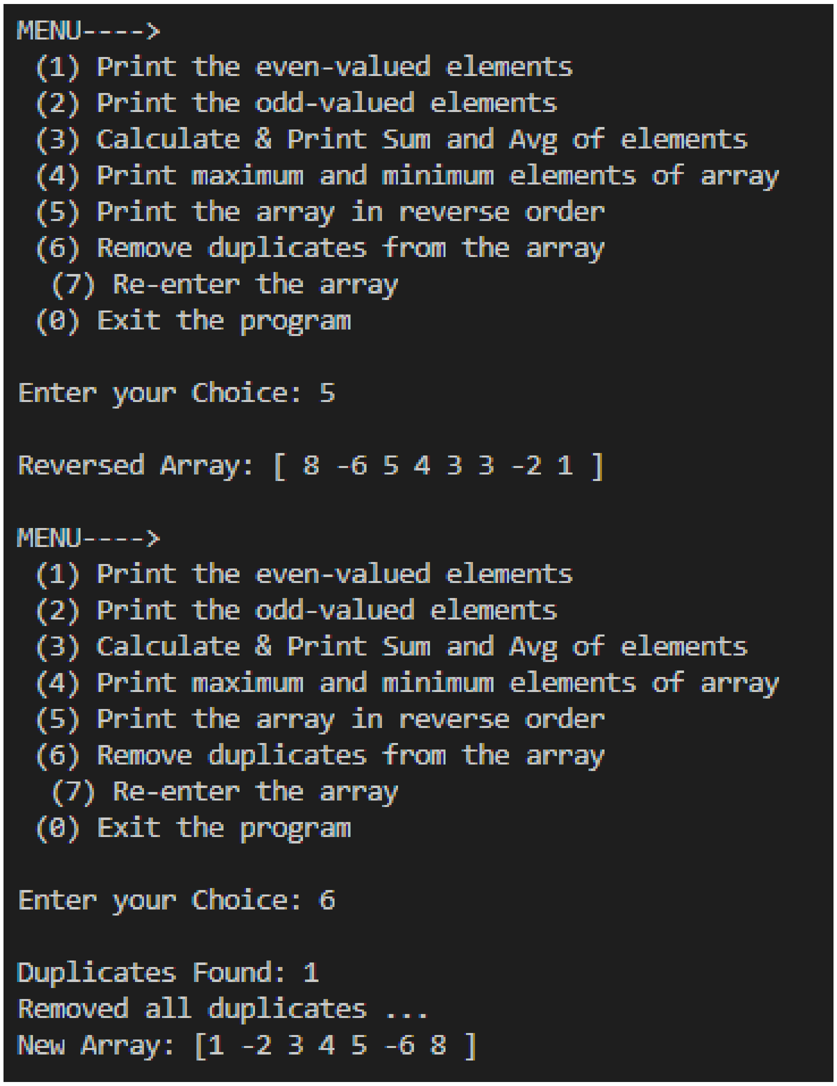
```
C:\Users\DELL\Desktop\C++ Practical>g++ menu.cpp -o menuKhushal

C:\Users\DELL\Desktop\C++ Practical>menuKhushal
Enter Size of the Array: 8
Enter elements of the array: 1 -2 3 3 4 5 -6 8
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 1

Even elements are: { -2 4 -6 8 }

MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 2

Odd elements are: { 1 3 3 5 }
```



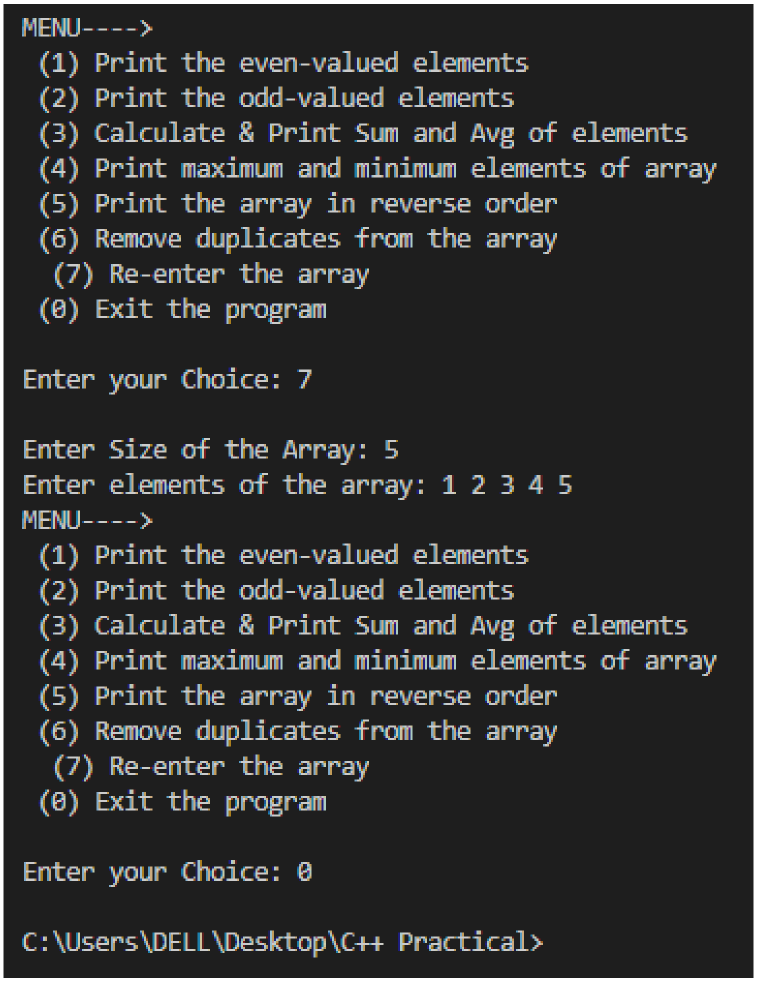
```
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 3

Sum of elements: 16
Average of elements: 2
```



```
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 4

Minimum= -6
Maximum= 8
```



```
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 5

Reversed Array: [ 8 -6 5 4 3 3 -2 1 ]

MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 6

Duplicates Found: 1
Removed all duplicates ...
New Array: [1 -2 3 4 5 -6 8 ]
```



```
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 7

Enter Size of the Array: 5
Enter elements of the array: 1 2 3 4 5
MENU---->
 (1) Print the even-valued elements
 (2) Print the odd-valued elements
 (3) Calculate & Print Sum and Avg of elements
 (4) Print maximum and minimum elements of array
 (5) Print the array in reverse order
 (6) Remove duplicates from the array
  (7) Re-enter the array
 (0) Exit the program

Enter your Choice: 0

C:\Users\DELL\Desktop\C++ Practical>
```

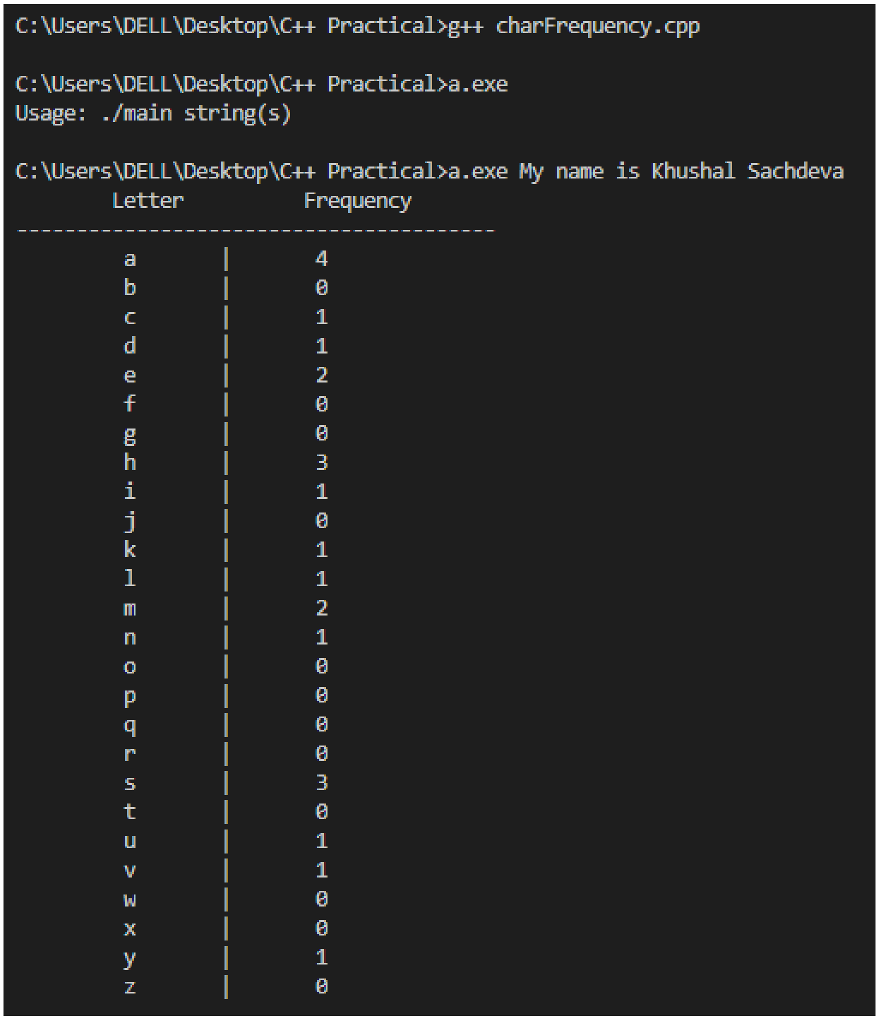
Practical - 10

Objective : Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments

Code :

```
/*
QUESTION-10: Write a program that prints a table indicating the number of occurrences
of each alphabet in the text entered as command line arguments
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <iomanip>
#include <cctype>
#define FREQ_SIZE 26
using namespace std;
int main(int argc, char **argv)
{
    // ensure proper usage
    if (argc == 1)
    {
        cerr << "Usage: ./main string(s)\n";
        return 1;
    }
    // initialise array containing frequencies of letters
    int frequency[FREQ_SIZE] = {0};
    // iterate over all the command-line arguments passed
    for (int i = 1; i < argc; i++)
        // iterate over each character in a word
        for (int j = 0; argv[i][j] != '\0'; j++)
            // handle only alphabetic letters
            if (isalpha(argv[i][j]))
            {
                // convert each letter to lowercase
                argv[i][j] = tolower(argv[i][j]);
                // increment frequency of each letter in word
                frequency[(int)(argv[i][j] - 'a')]++;}
    // print the header of the table
    cout << "\tLetter\t\tFrequency\n";
    cout << setw(40) << setfill('-') << "" << endl;
    // print the letter and its number of occurrences
    for (int i = 0; i < FREQ_SIZE; i++)
    {
        cout << "\t ";
        cout << setw(7) << setfill(' ') << left << (char)('a' + i);
        cout << " | ";
        cout << setw(7) << setfill(' ') << right << frequency[i] << endl;
    }
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ charFrequency.cpp

C:\Users\DELL\Desktop\C++ Practical>a.exe
Usage: ./main string(s)

C:\Users\DELL\Desktop\C++ Practical>a.exe My name is Khushal Sachdeva
        Letter          Frequency
----------------------------------------
         a       |       4
         b       |       0
         c       |       1
         d       |       1
         e       |       2
         f       |       0
         g       |       0
         h       |       3
         i       |       1
         j       |       0
         k       |       1
         l       |       1
         m       |       2
         n       |       1
         o       |       0
         p       |       0
         q       |       0
         r       |       0
         s       |       3
         t       |       0
         u       |       1
         v       |       1
         w       |       0
         x       |       0
         y       |       1
         z       |       0
```

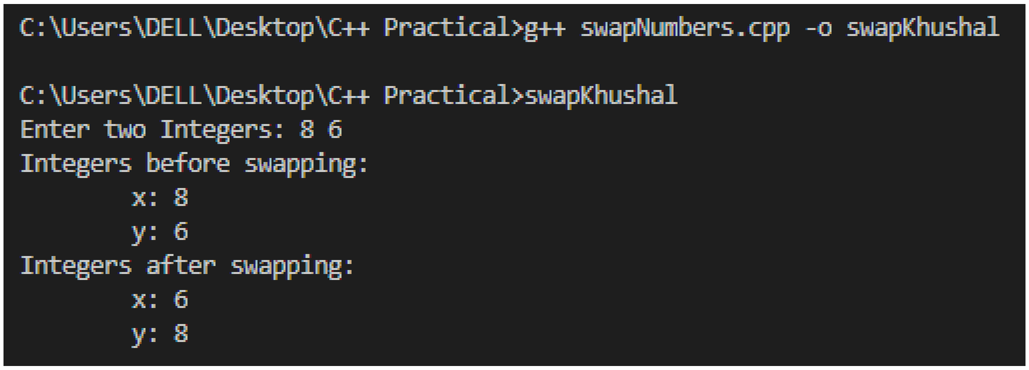
Practical - 11

Objective : Write a program that swaps two numbers using pointers.

Code :

```
/*
QUESTION-11: Write a program that swaps two numbers using pointers.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;

void swapPointer(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int x, y;
    cout << "Enter two Integers: ";
    cin >> x >> y;
    cout << "Integers before swapping: " << endl
         << "\tx: " << x << endl
         << "\ty: " << y << endl;
    swapPointer(&x, &y);
    cout << "Integers after swapping: " << endl
         << "\tx: " << x << endl
         << "\ty: " << y << endl;
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ swapNumbers.cpp -o swapKhushal

C:\Users\DELL\Desktop\C++ Practical>swapKhushal
Enter two Integers: 8 6
Integers before swapping:
        x: 8
        y: 6
Integers after swapping:
        x: 6
        y: 8
```

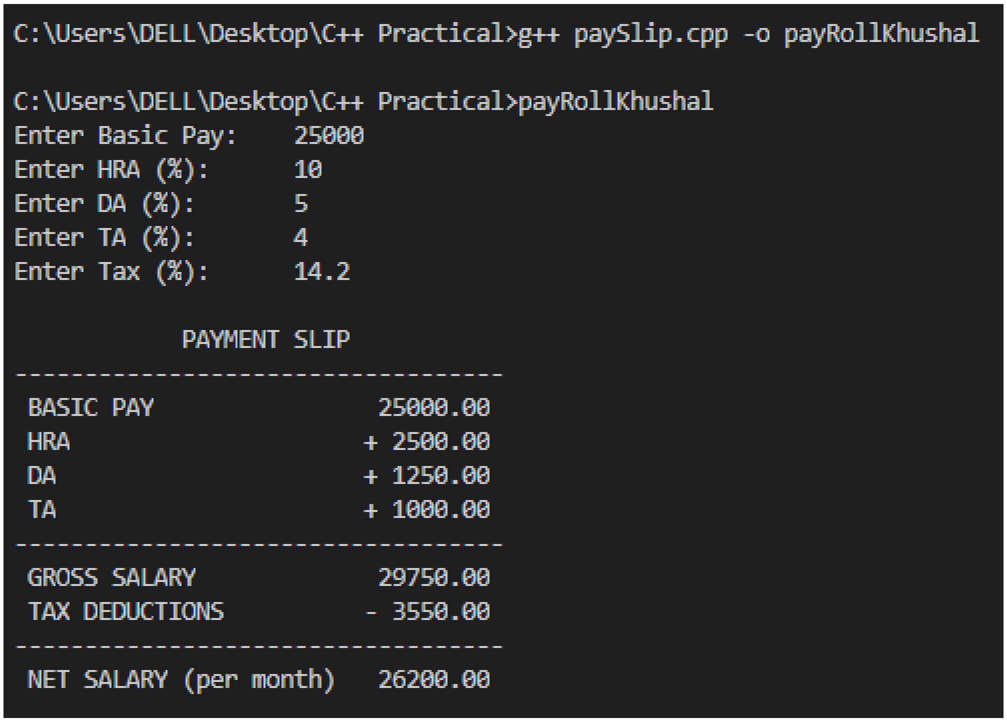
Practical - 12

Objective : Write a program to generate pay-slip of salaried employees.

Code :

```
/*
QUESTION-12: Write a program to generate pay-slip of salaried employees.
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    float basic, hra, da, ta, tax;
    cout << left;
    cout << setw(20) << "Enter Basic Pay:";
    cin >> basic;
    cout << setw(20) << "Enter HRA (%):";
    cin >> hra;
    cout << setw(20) << "Enter DA (%):";
    cin >> da;
    cout << setw(20) << "Enter TA (%):";
    cin >> ta;
    cout << setw(20) << "Enter Tax (%):";
    cin >> tax;
    hra = basic * (hra / 100);
    da = basic * (da / 100);
    ta = basic * (ta / 100);
    tax = basic * (tax / 100);
    cout << fixed << showpoint << endl;
    cout << right << setw(24) << "PAYMENT SLIP" << endl;
    cout << left << setw(35) << setfill('-') << "" << endl;
    cout << left << setw(25) << setfill(' ') << " BASIC PAY";
    cout << right << setw(9) << setprecision(2) << basic << endl;
    cout << left << setw(25) << setfill(' ') << " HRA";
    cout << '+';
    cout << right << setw(8) << setprecision(2) << hra << endl;
    cout << left << setw(25) << setfill(' ') << " DA";
    cout << '+';
    cout << right << setw(8) << setprecision(2) << da << endl;
    cout << left << setw(25) << setfill(' ') << " TA";
    cout << '+';
    cout << right << setw(8) << setprecision(2) << ta << endl;
    cout << left << setw(35) << setfill('-') << "" << endl;
    cout << left << setw(25) << setfill(' ') << " GROSS SALARY";
    cout << right << setw(9) << setprecision(2) << basic + ta + da + hra
         << endl;
    cout << left << setw(25) << setfill(' ') << " TAX DEDUCTIONS";
    cout << '-';
    cout << right << setw(8) << setprecision(2) << tax << endl;
    cout << left << setw(35) << setfill('-') << "" << endl;
    cout << left << setw(25) << setfill(' ') << " NET SALARY (per month)";
    cout << right << setw(9) << setprecision(2) << basic + ta + da + hra -
         tax << endl;
    cout << endl;
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ paySlip.cpp -o payRollKhushal

C:\Users\DELL\Desktop\C++ Practical>payRollKhushal
Enter Basic Pay:    25000
Enter HRA (%):      10
Enter DA (%):       5
Enter TA (%):       4
Enter Tax (%):      14.2

            PAYMENT SLIP
-----------------------------------
 BASIC PAY                25000.00
 HRA                     + 2500.00
 DA                      + 1250.00
 TA                      + 1000.00
-----------------------------------
 GROSS SALARY             29750.00
 TAX DEDUCTIONS          - 3550.00
-----------------------------------
 NET SALARY (per month)   26200.00
```

Practical - 13

Objective : Write a menu driven program to perform following operations on strings:

(g) Show address of each character in string.
(h) Concatenate two strings without using strcat function.
(i) Concatenate two strings using strcat function.
(j) Compare two strings.
(k) Calculate the length of the string (use pointers).
(l) Convert all lowercase characters to uppercase.
(m) Convert all uppercase characters to lowercase.
(n) Calculate the number of vowels.
(o) Reverse the string.

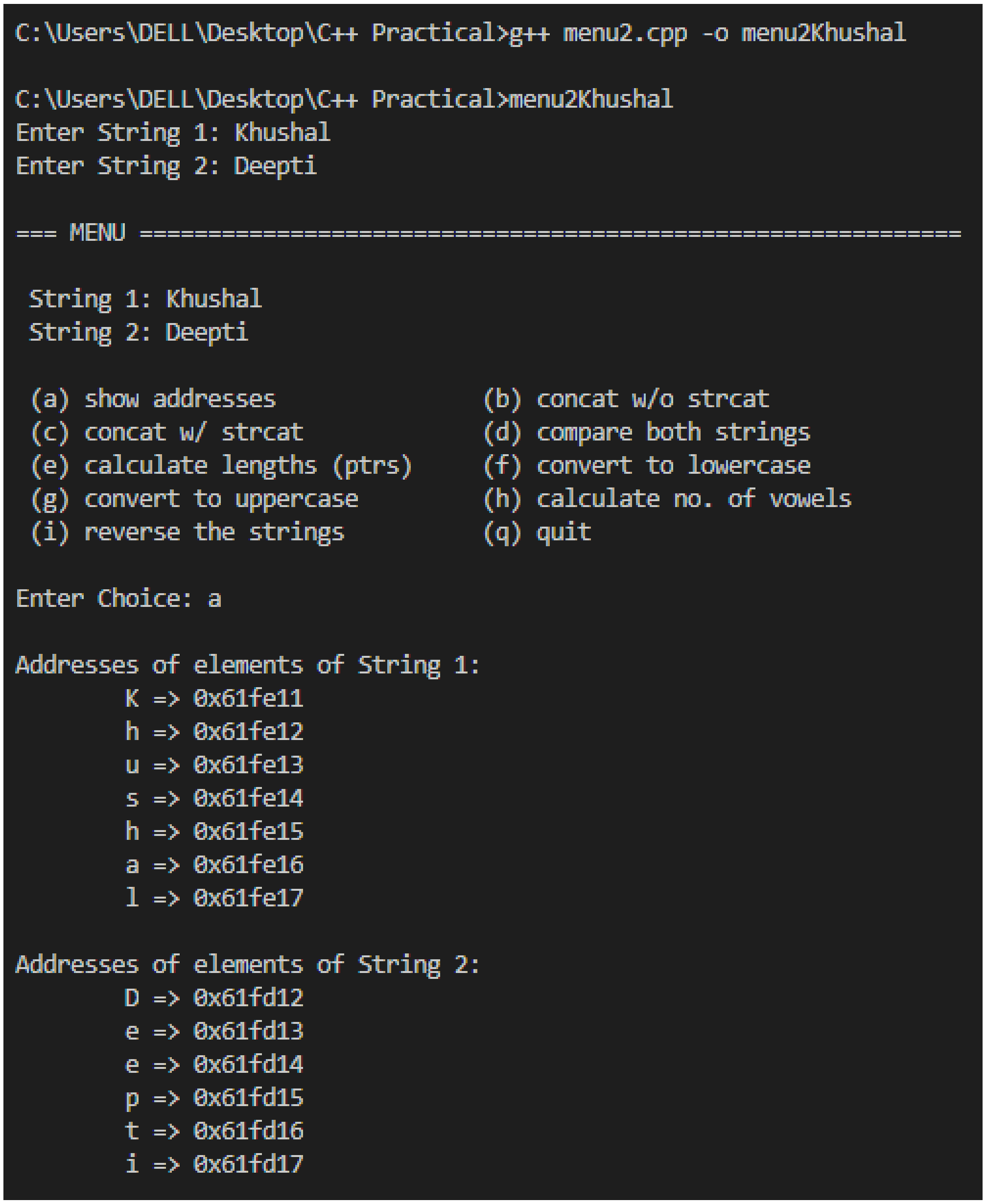
Code :

```
/*
QUESTION-13: Write a menu driven program to perform following operations on strings:
(g) Show address of each character in string.
(h) Concatenate two strings without using strcat function.
(i) Concatenate two strings using strcat function.
(j) Compare two strings.
(k) Calculate the length of the string (use pointers).
(l) Convert all lowercase characters to uppercase.
(m) Convert all uppercase characters to lowercase.
(n) Calculate the number of vowels.
(o) Reverse the string.
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <cctype>
using namespace std;
int stringLength(char[]);
int countVowels(char[]);
```

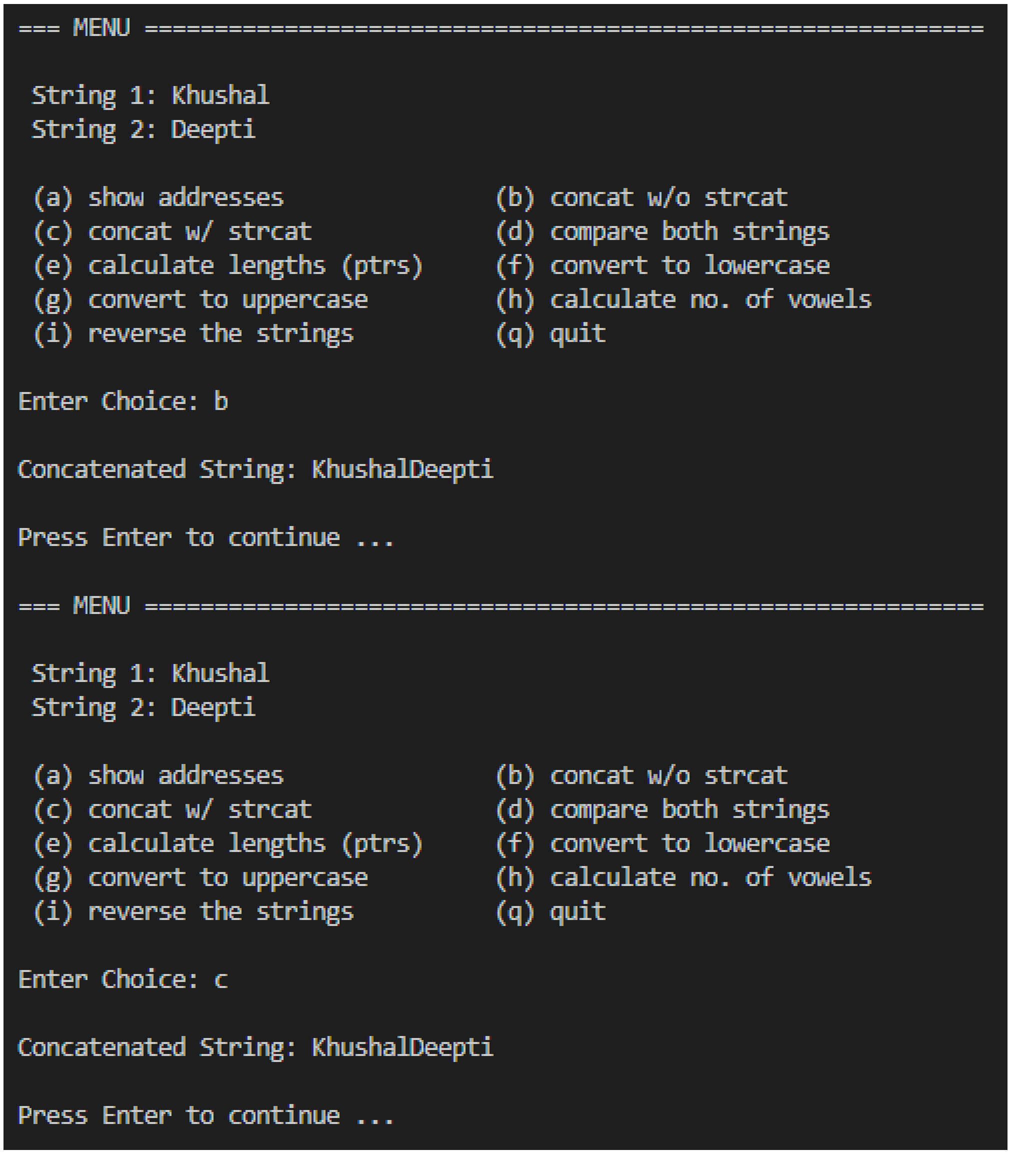
```
void showAddresses(char[]);
void showMenu(char[], char[]);
char *lowercase(char[], char[]);
char *uppercase(char[], char[]);
char *reverseString(char[], char[]);
void compareStrings(char[], char[]);
char *concatWithStrcat(char[], char[], char[]);
char *concatWithoutStrcat(char[], char[], char[]);
int main()
{
    char string1[255], string2[255], choice = ' ';
    cout << "Enter String 1: ";
    cin >> string1;
    cout << "Enter String 2: ";
    cin >> string2;
    while (choice != 'q')
    {
        showMenu(string1, string2);
        cout << "Enter Choice: ";
        cin >> choice;
        switch (choice)
        {
        case 'a':
            cout << endl;
            cout << "Addresses of elements of String 1:\n";
            showAddresses(string1);
            cout << endl;
            cout << "Addresses of elements of String 2:\n";
            showAddresses(string2);
            cout << endl;
            break;
        case 'b':
            cout << endl;
            {
                char t[255];
                cout << "Concatenated String: ";
                concatWithoutStrcat(string1, string2, t);
                cout << t << endl;
            }
            cout << endl;
            break;
        case 'c':
            cout << endl;
            {
                char t[255];
                cout << "Concatenated String: ";
                concatWithStrcat(string1, string2, t);
                cout << t << endl;
            }
            cout << endl;
            break;
        case 'd':
            cout << endl;
            cout << "Result of Comparison: ";
            compareStrings(string1, string2);
            cout << endl;
            break;
        case 'e':
            cout << "\nLength of String 1: " << stringLength(string1);
            cout << "\nLength of String 2: " << stringLength(string2);
            cout << endl
                 << endl;
            break;
        case 'f':
        {
            char t[255];
            cout << "\nString 1 in Lowercase: " << lowercase(string1, t);
            cout << "\nString 2 in Lowercase: " << lowercase(string2, t);
        }
            cout << endl
                 << endl;
            break;
        case 'g':
        {
```

```
            char t[255];
            cout << "\nString 1 in Uppercase: " << uppercase(string1, t);
            cout << "\nString 2 in Uppercase: " << uppercase(string2, t);
        }
            cout << endl
                 << endl;
            break;
        case 'h':
            cout << "\nVowels in String 1: " << countVowels(string1);
            cout << "\nVowels in String 2: " << countVowels(string2);
            cout << endl
                 << endl;
            break;
        case 'i':
        {
            char t[255];
            cout << "\nReverse of String 1: " << reverseString(string1, t);
            cout << "\nReverse of String 2: " << reverseString(string2, t);
        }
            cout << endl
                 << endl;
            break;
        default:
            break;
        }
        if (choice != 'q')
        {
            cin.get();
            cout << "Press Enter to continue ...";
            cin.get();
        }
    };
    return 0;
}
void showMenu(char a[], char b[])
{
    cout << endl;
    cout << setw(3) << setfill('=') << "";
    cout << " MENU ";
    cout << setw(60) << "";
    cout << left;
    cout << endl
         << endl;
    cout << " String 1: " << a;
    cout << "\n String 2: " << b;
    cout << endl
         << endl;
    cout << setw(34) << setfill(' ') << " (a) show addresses";
    cout << setw(34) << "(b) concat w/o strcat";
    cout << endl;
    cout << setw(34) << " (c) concat w/ strcat";
    cout << setw(34) << "(d) compare both strings";
    cout << endl;
    cout << setw(34) << " (e) calculate lengths (ptrs)";
    cout << setw(34) << "(f) convert to lowercase";
    cout << endl;
    cout << setw(34) << " (g) convert to uppercase";
    cout << setw(34) << "(h) calculate no. of vowels";
    cout << endl;
    cout << setw(34) << " (i) reverse the strings";
    cout << setw(34) << "(q) quit";
    cout << endl
         << endl;
    return;
}
void showAddresses(char a[])
{
    for (int i = 0; i < stringLength(a); i++)
        cout << '\t' << a[i] << " => " << (void *)&a[i] << endl;
    return;
}
char *concatWithoutStrcat(char a[], char b[], char t[])
{
    int i = stringLength(a), j;
    strcpy(t, a);
    for (j = 0; j < stringLength(b); j++, i++)
        t[i] = b[j];
    t[i] = '\0';
    return t;
}
char *concatWithStrcat(char a[], char b[], char t[])
{
    strcpy(t, a);
    strcat(t, b);
    return t;
}
void compareStrings(char a[], char b[])
{
    if (stringLength(a) == stringLength(b))
        cout << "length(String 1) == length(String 2)" << endl;
    else if (stringLength(a) > stringLength(b))
        cout << "length(String 1) > length(String 2)" << endl;
    else if (stringLength(a) < stringLength(b))
        cout << "length(String 1) < length(String 2)" << endl;
    int flag = 0;
    for (int i = 0, j = 0; i < stringLength(a); i++, j++)
        if (a[i] == b[j])
            flag++;
    if (flag == stringLength(a))
        cout << "String 1 is the same as String 2!" << endl;
    return;
}
int stringLength(char *a)
{
    int i = 0;
    for (char *c = a; *c != '\0'; c++, i++)
        ;
    return i;
}
char *lowercase(char a[], char t[])
{
    strcpy(t, a);
    for (int i = 0; i < stringLength(t); i++)
        if (t[i] >= 'A' && t[i] <= 'Z')
            t[i] = t[i] + 32;
    return t;
}
char *uppercase(char a[], char t[])
{
    strcpy(t, a);
    for (int i = 0; i < stringLength(t); i++)
        if (t[i] >= 'a' && t[i] <= 'z')
            t[i] = t[i] - 32;
    return t;
}
int countVowels(char a[])
{
    int count = 0;
    char t[255];
    for (int i = 0; i < stringLength(a); i++)
        if (lowercase(a, t)[i] == 'a' || lowercase(a, t)[i] == 'e' || lowercase(a, t)[i]
== 'i' || lowercase(a, t)[i] == 'o' || lowercase(a, t)[i] == 'u')
            count++;
    return count;
}
char *reverseString(char a[], char t[])
{
    strcpy(t, a);
    int i = 0, j = stringLength(t) - 1, temp;
    while (i < j)
    {
        temp = t[i];
        t[i] = t[j];
        t[j] = temp;
        i++;
        j--;
    }
    return t;}
```

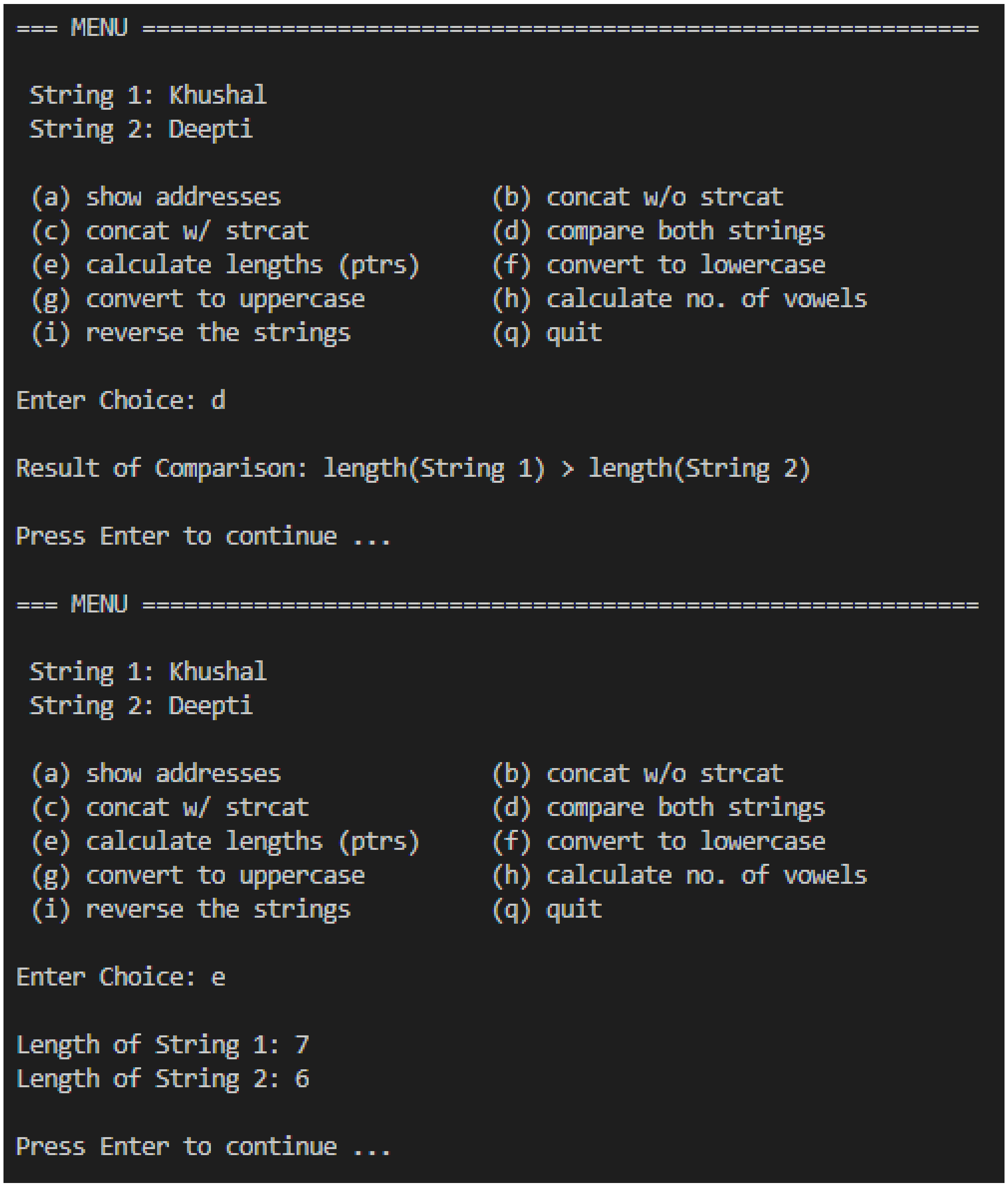
Output :



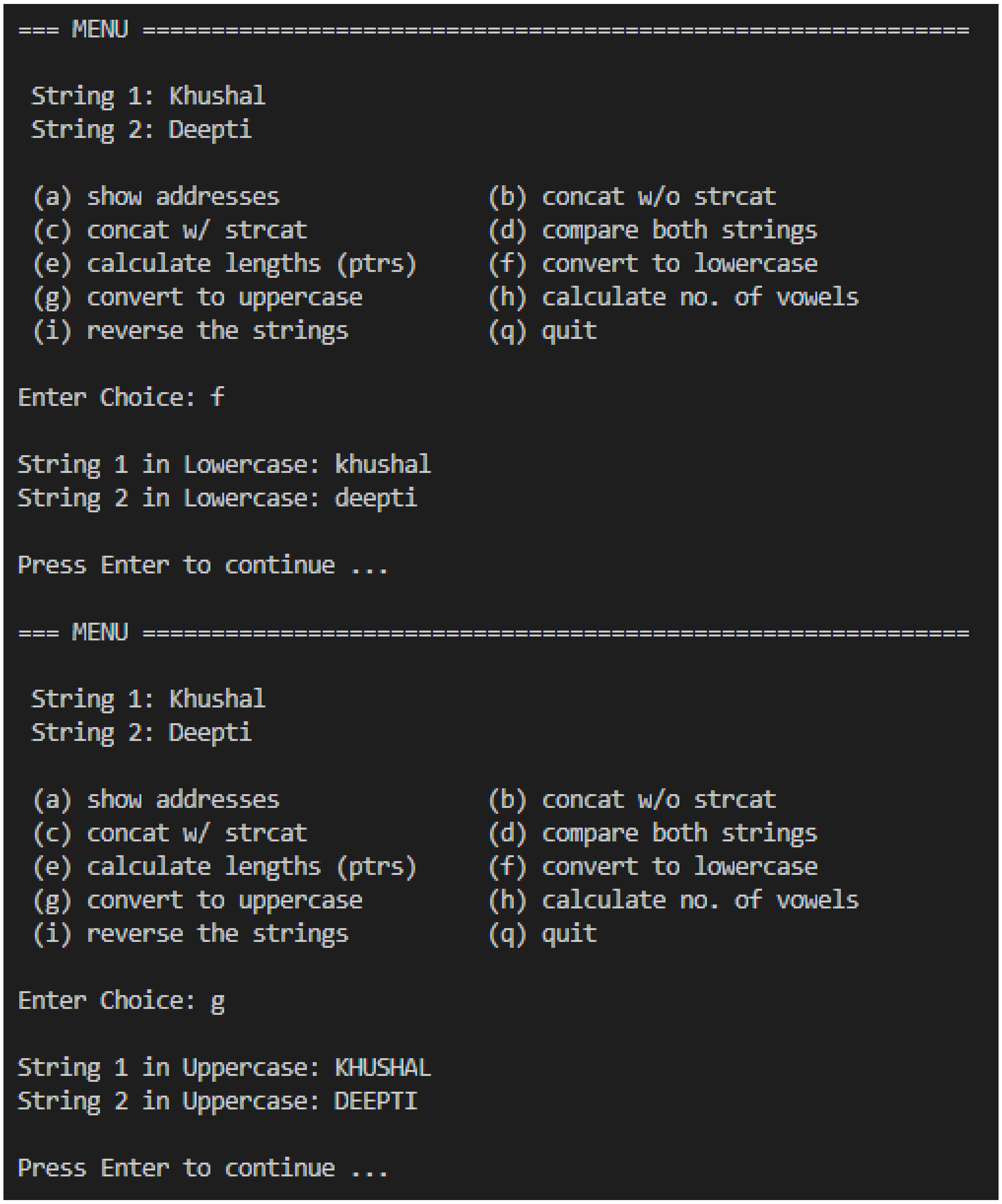
```
C:\Users\DELL\Desktop\C++ Practical>g++ menu2.cpp -o menu2Khushal

C:\Users\DELL\Desktop\C++ Practical>menu2Khushal
Enter String 1: Khushal
Enter String 2: Deepti

=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: a

Addresses of elements of String 1:
        K => 0x61fe11
        h => 0x61fe12
        u => 0x61fe13
        s => 0x61fe14
        h => 0x61fe15
        a => 0x61fe16
        l => 0x61fe17

Addresses of elements of String 2:
        D => 0x61fd12
        e => 0x61fd13
        e => 0x61fd14
        p => 0x61fd15
        t => 0x61fd16
        i => 0x61fd17
```



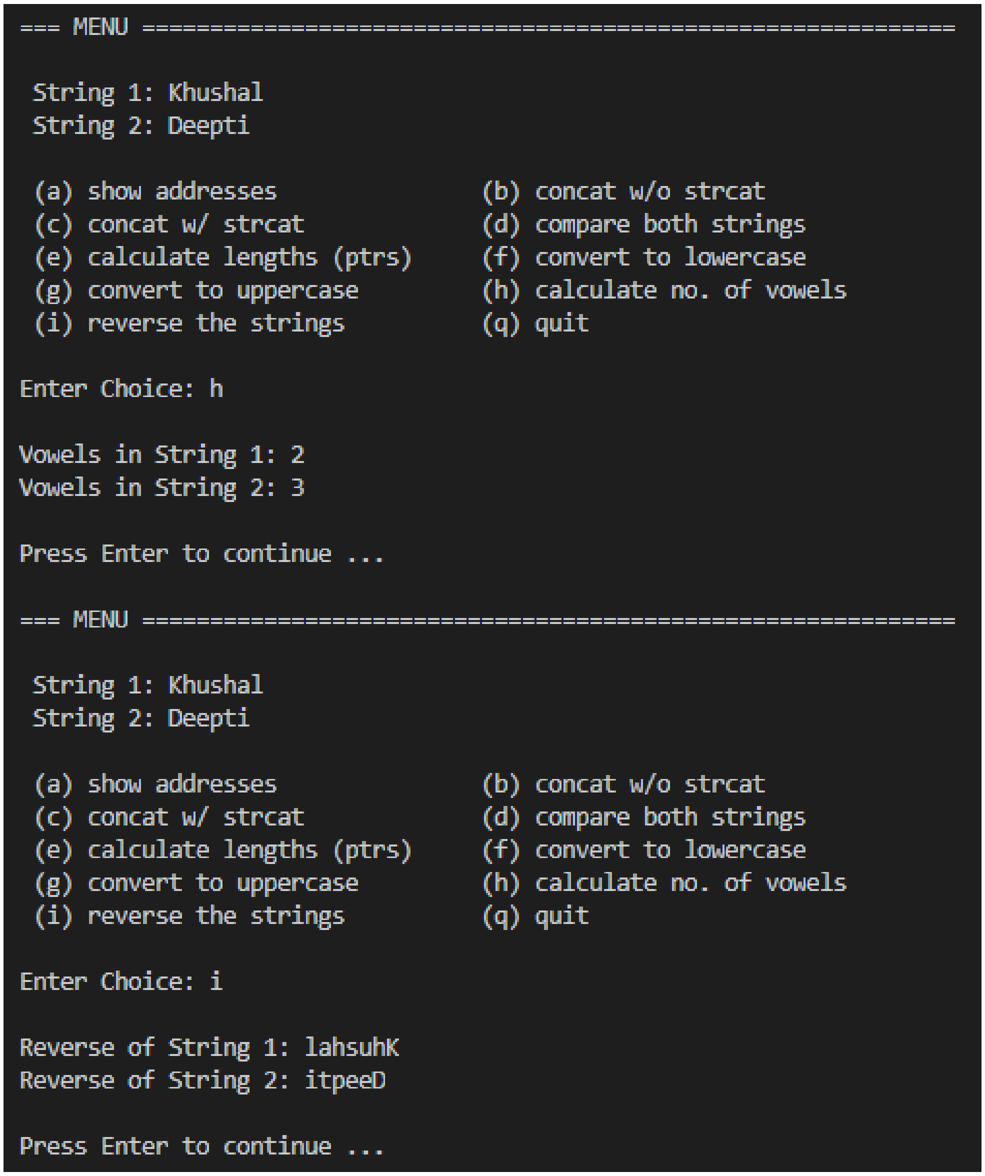
```
=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: b

Concatenated String: KhushalDeepti

Press Enter to continue ...

=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: c

Concatenated String: KhushalDeepti

Press Enter to continue ...
```



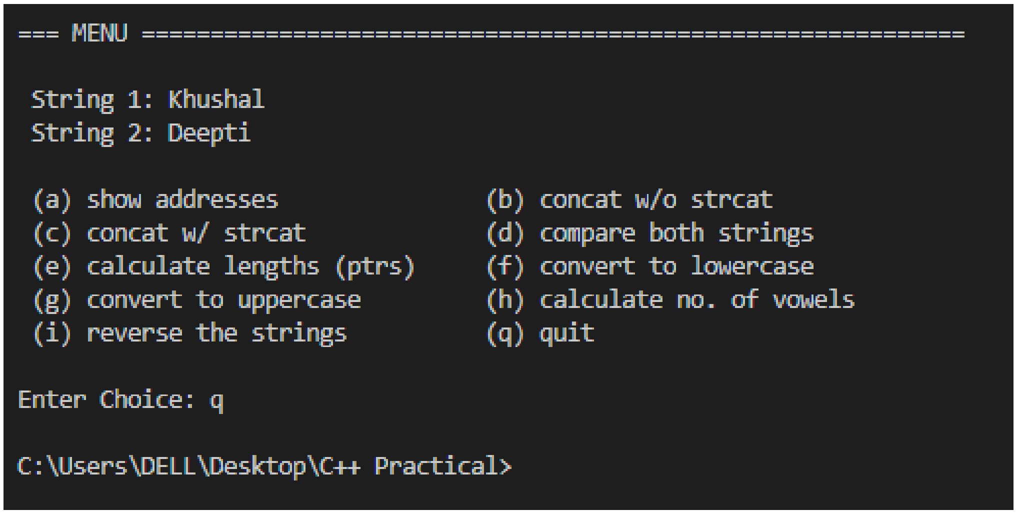
```
=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: d

Result of Comparison: length(String 1) > length(String 2)

Press Enter to continue ...

=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: e

Length of String 1: 7
Length of String 2: 6

Press Enter to continue ...
```



```
=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses              (b) concat w/o strcat
 (c) concat w/ strcat            (d) compare both strings
 (e) calculate lengths (ptrs)    (f) convert to lowercase
 (g) convert to uppercase        (h) calculate no. of vowels
 (i) reverse the strings         (q) quit

Enter Choice: f

String 1 in Lowercase: khushal
String 2 in Lowercase: deepti

Press Enter to continue ...

=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses              (b) concat w/o strcat
 (c) concat w/ strcat            (d) compare both strings
 (e) calculate lengths (ptrs)    (f) convert to lowercase
 (g) convert to uppercase        (h) calculate no. of vowels
 (i) reverse the strings         (q) quit

Enter Choice: g

String 1 in Uppercase: KHUSHAL
String 2 in Uppercase: DEEPTI

Press Enter to continue ...
```



```
=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: h

Vowels in String 1: 2
Vowels in String 2: 3

Press Enter to continue ...

=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: i

Reverse of String 1: lahsuhK
Reverse of String 2: itpeeD

Press Enter to continue ...
```



```
=== MENU ============================================================

 String 1: Khushal
 String 2: Deepti

 (a) show addresses               (b) concat w/o strcat
 (c) concat w/ strcat             (d) compare both strings
 (e) calculate lengths (ptrs)     (f) convert to lowercase
 (g) convert to uppercase         (h) calculate no. of vowels
 (i) reverse the strings          (q) quit

Enter Choice: q

C:\Users\DELL\Desktop\C++ Practical>
```

Practical - 14

Objective : Write a program using iteration to:

(a) Display the Fibonacci Series

(b) Calculate Factorial of a Number

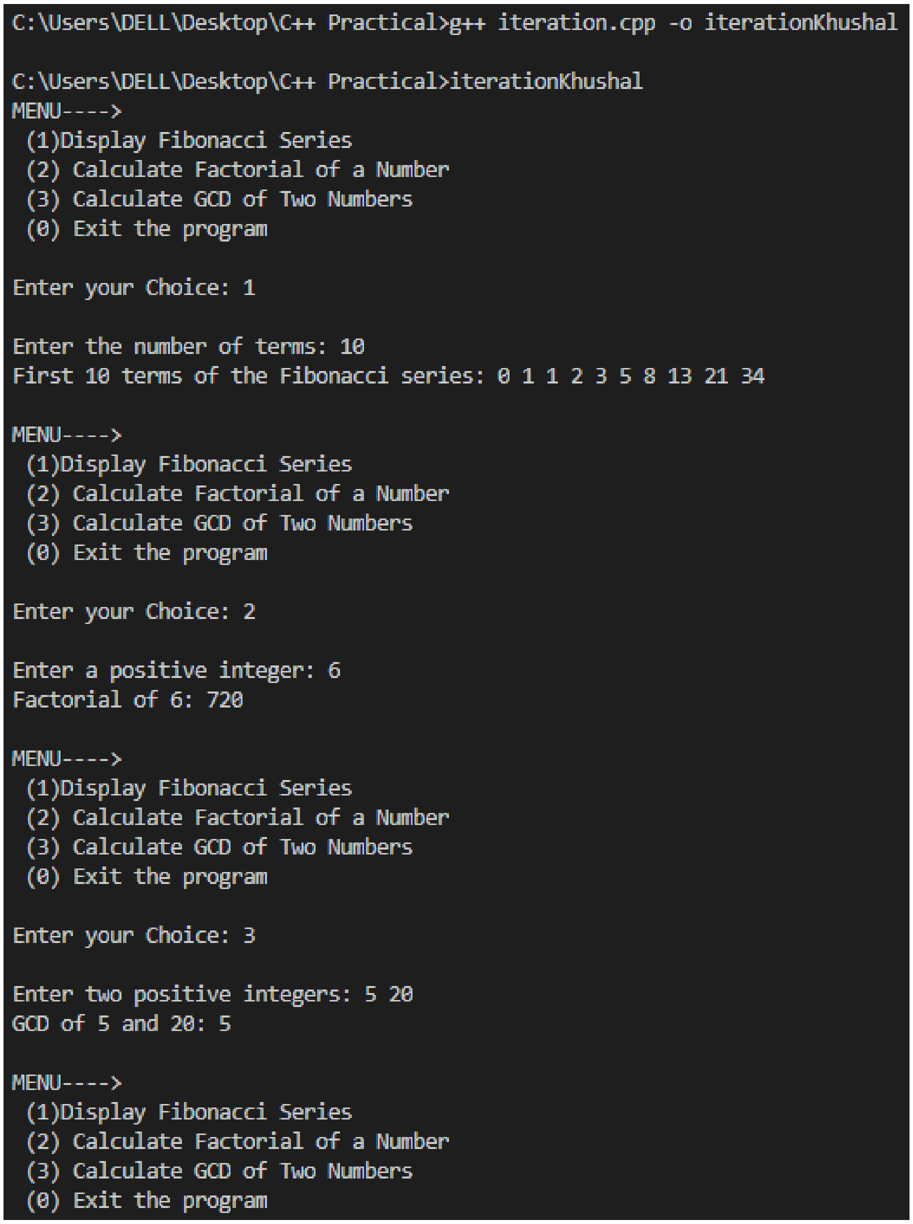
(c) Calculate GCD of Two Numbers

Code :

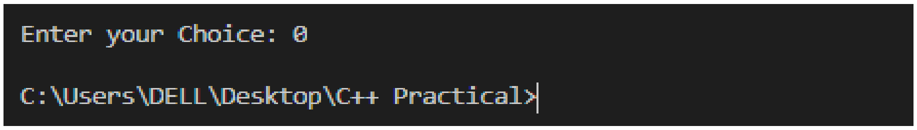
```
/*
QUESTION-14: Write a program using iteration to:
(a) Display the Fibonacci Series
(b) Calculate Factorial of a Number
(c) Calculate GCD of Two Numbers
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
int main()
{
    int choice;
    do{
        cout << "MENU---->" << endl;
        cout << " (1)Display Fibonacci Series\n"
             << " (2) Calculate Factorial of a Number\n"
             << " (3) Calculate GCD of Two Numbers\n"
             << " (0) Exit the program\n";
        cout << "\nEnter your Choice: ";
        cin >> choice;
        switch (choice){
        case (1):
        {
            int n;
            cout << "\nEnter the number of terms: ";
            cin >> n;
            cout << "First " << n << " terms of the Fibonacci series: ";
            for (int i = 0, next, first = 0, second = 1; i < n; i++){
                if (i <= 1)
                    next = i;
                else{
                    next = first + second;
                    first = second;
                    second = next;
                }
                cout << next << " ";
            }
            cout << endl
                 << endl;
            break;
        }
        case (2):
        {
            int n;
            cout << "\nEnter a positive integer: ";
            cin >> n;
            int product = 1;
            for (int i = n; i > 0; i--)
                product *= i;
            cout << "Factorial of " << n << ": " << product << endl
                 << endl;
            break;
        }
        case (3):
        {
            int x, y, gcd;
            cout << "\nEnter two positive integers: ";
            cin >> x >> y;
            for (int i = 1; i <= x && i <= y; i++)
                if (x % i == 0 && y % i == 0)
                    gcd = i;
            cout << "GCD of " << x << " and " << y << ": " << gcd << endl
                 << endl;
            break;
        }
        default:
```

```
            break;
        }
    } while (choice != 0);
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ iteration.cpp -o iterationKhushal

C:\Users\DELL\Desktop\C++ Practical>iterationKhushal
MENU---->
 (1)Display Fibonacci Series
 (2) Calculate Factorial of a Number
 (3) Calculate GCD of Two Numbers
 (0) Exit the program

Enter your Choice: 1

Enter the number of terms: 10
First 10 terms of the Fibonacci series: 0 1 1 2 3 5 8 13 21 34

MENU---->
 (1)Display Fibonacci Series
 (2) Calculate Factorial of a Number
 (3) Calculate GCD of Two Numbers
 (0) Exit the program

Enter your Choice: 2

Enter a positive integer: 6
Factorial of 6: 720

MENU---->
 (1)Display Fibonacci Series
 (2) Calculate Factorial of a Number
 (3) Calculate GCD of Two Numbers
 (0) Exit the program

Enter your Choice: 3

Enter two positive integers: 5 20
GCD of 5 and 20: 5

MENU---->
 (1)Display Fibonacci Series
 (2) Calculate Factorial of a Number
 (3) Calculate GCD of Two Numbers
 (0) Exit the program
```



```
Enter your Choice: 0

C:\Users\DELL\Desktop\C++ Practical>
```

Practical - 15

Objective : Create Matrix class. Write a menu driven program to perform following Matrix operations:

(d) Sum

(e) Difference

(f) Product

(g) Transpose

Code :

```
/*
QUESTION-15: Create Matrix class. Write a menu driven program to perform following
Matrix operations:
(d) Sum
(e) Difference
(f) Product
(g) Transpose
Written By: Khushal Sachdeva
*/
#include <iostream>
#define MAX_DIM 100
using namespace std;

class Matrix
{
private:
    int rows;
    int cols;
    int matrix[MAX_DIM][MAX_DIM];

public:
    Matrix(int, int);
    void get();
    void put();
    void sum(Matrix &);
    void difference(Matrix &);
    void product(Matrix &);
    void transpose();
```

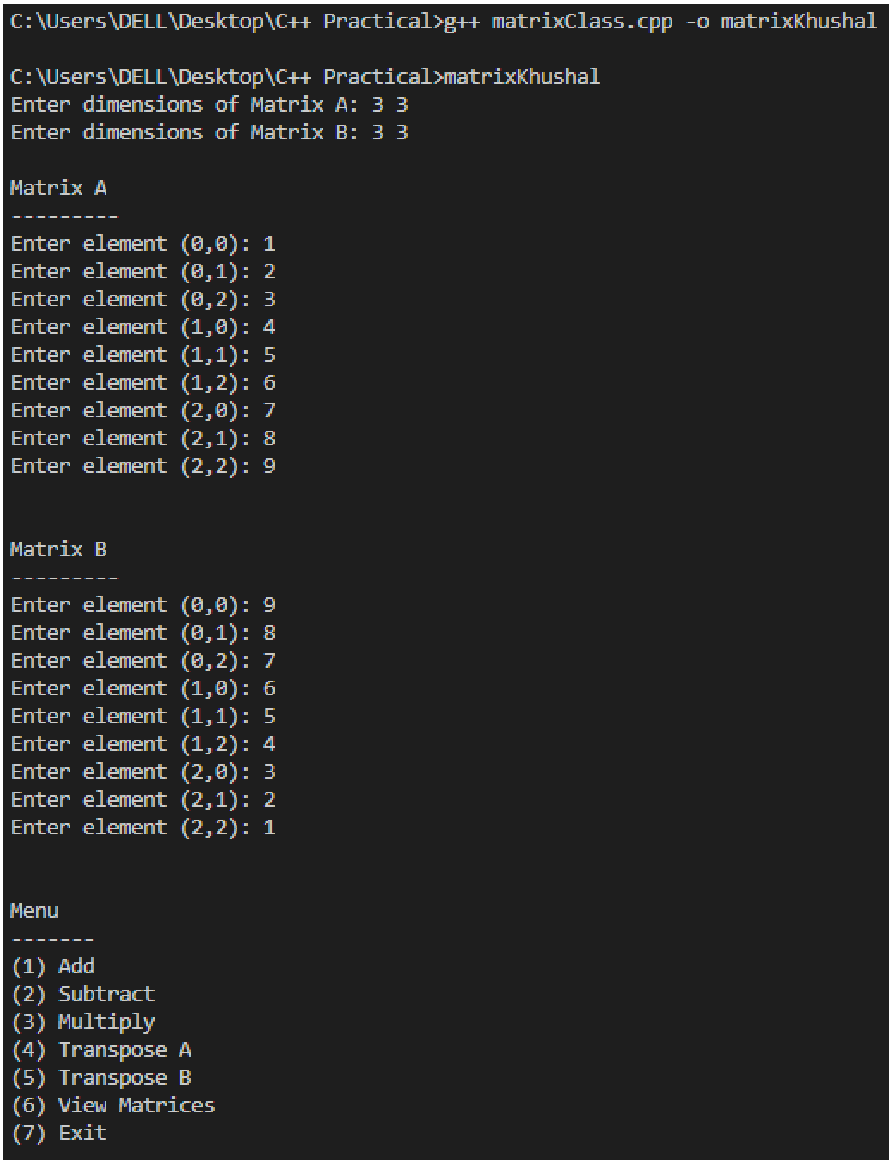
```
};
Matrix::Matrix(int r = 3, int c = 3)
{
    rows = r;
    cols = c;
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            matrix[i][j] = 0;
}
void Matrix::get()
{
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
        {
            cout << "Enter element (" << i << "," << j << "): ";
            cin >> matrix[i][j];
        }
    return;
}
void Matrix::put()
{
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
            cout << matrix[i][j] << "\t";
        cout << endl;
    }
}
void Matrix::sum(Matrix &o)
{
    if (rows != o.rows || cols != o.cols)
    {
        cout << "Matrices cannot be added." << endl;
        return;
    }
    Matrix temp(rows, cols);
    cout << "SUM:\n\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            temp.matrix[i][j] = matrix[i][j] + o.matrix[i][j];
    temp.put();
    return;
}
void Matrix::difference(Matrix &o)
{
    if (rows != o.rows || cols != o.cols)
    {
        cout << "Matrices cannot be subtracted." << endl;
        return;
    }
    Matrix temp(rows, cols);
    cout << "DIFFERENCE:\n\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            temp.matrix[i][j] = matrix[i][j] - o.matrix[i][j];
    temp.put();
    return;
}
void Matrix::product(Matrix &o)
{
    if (cols != o.rows)
    {
        cout << "Matrices cannot be multiplied." << endl;
        return;
    }
    Matrix temp(rows, o.cols);
    cout << "PRODUCT:\n\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            for (int k = 0; k < o.rows; k++)
                temp.matrix[i][j] += matrix[i][k] * o.matrix[k][j];
    temp.put();
    return;
}
```

```cpp
void Matrix::transpose()
{
    Matrix temp(cols, rows);
    cout << "TRANSPOSE:\n\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            temp.matrix[j][i] = matrix[i][j];
    temp.put();
    return;
}
void getInputs(Matrix &, Matrix &);
void handleMenu(Matrix &, Matrix &);
int main()
{
    int r1, c1, r2, c2;
    cout << "Enter dimensions of Matrix A: ";
    cin >> r1 >> c1;
    cout << "Enter dimensions of Matrix B: ";
    cin >> r2 >> c2;
    Matrix A(r1, c1), B(r2, c2);
    getInputs(A, B);
    handleMenu(A, B);
    cout << "Exiting...\n";
    return 0;
}
void getInputs(Matrix &A, Matrix &B)
{
    cout << "\nMatrix A\n---------\n";
    A.get();
    cout << endl;
    cout << "\nMatrix B\n---------\n";
    B.get();
    cout << endl;
    return;
}
void handleMenu(Matrix &A, Matrix &B)
{
```

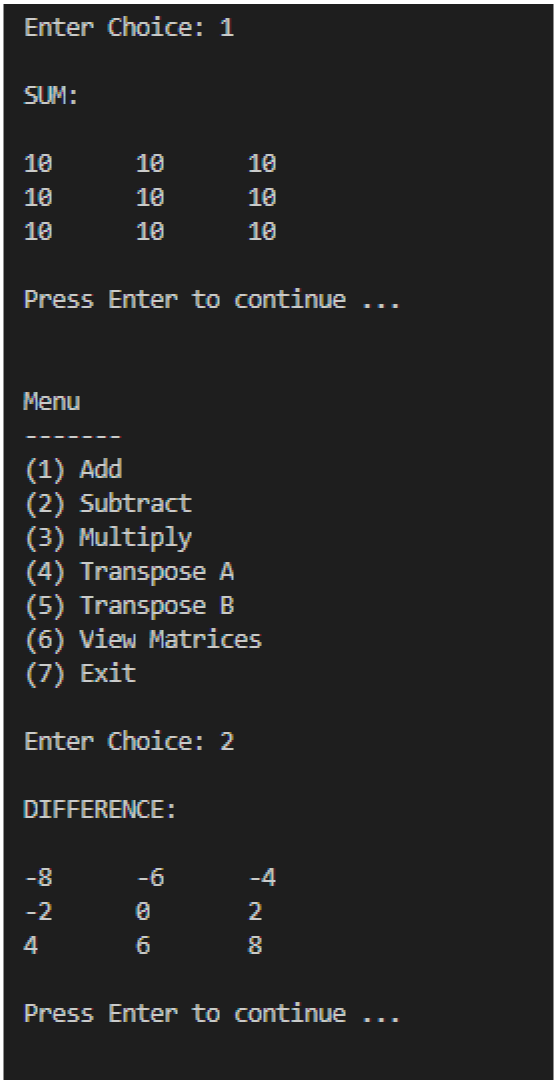
```
int ch = 0;
do
{
    cout << "\nMenu\n-------\n(1) Add\n(2) Subtract";
    cout << "\n(3) Multiply\n(4) Transpose A\n(5) Transpose B";
    cout << "\n(6) View Matrices\n(7) Exit\n\nEnter Choice: ";
    cin >> ch;
    cout << endl;
    switch (ch)
    {
    case 1:
        A.sum(B);
        break;
    case 2:
        A.difference(B);
        break;
    case 3:
        A.product(B);
        break;
    case 4:
        A.transpose();
        break;
    case 5:
        B.transpose();
        break;
    case 6:
        cout << "Matrix A\n---------\n";
        A.put();
        cout << endl;
        cout << "Matrix B\n---------\n";
        B.put();
        break;
    case 7:
    default:
        break;
    }
    if (ch != 7)
```

```
        {
            cout << "\nPress Enter to continue ...\n";
            cin.ignore();
            cin.get();
        }
    } while (ch != 7);
    return;
}
```

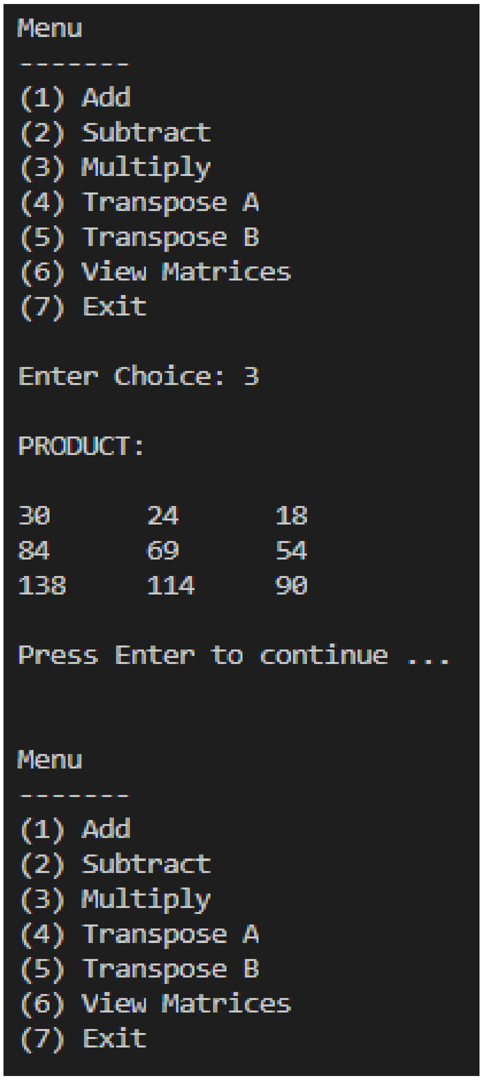
Output :



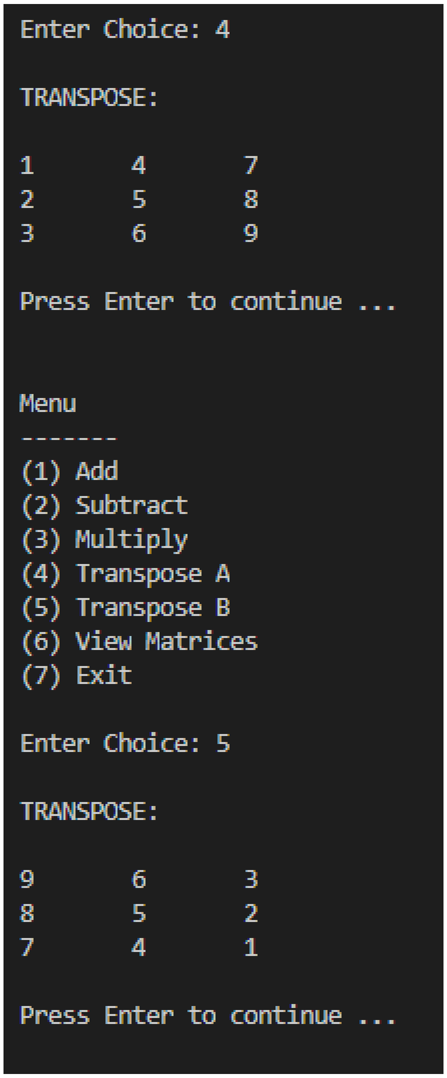
```
C:\Users\DELL\Desktop\C++ Practical>g++ matrixClass.cpp -o matrixKhushal

C:\Users\DELL\Desktop\C++ Practical>matrixKhushal
Enter dimensions of Matrix A: 3 3
Enter dimensions of Matrix B: 3 3

Matrix A
---------
Enter element (0,0): 1
Enter element (0,1): 2
Enter element (0,2): 3
Enter element (1,0): 4
Enter element (1,1): 5
Enter element (1,2): 6
Enter element (2,0): 7
Enter element (2,1): 8
Enter element (2,2): 9


Matrix B
---------
Enter element (0,0): 9
Enter element (0,1): 8
Enter element (0,2): 7
Enter element (1,0): 6
Enter element (1,1): 5
Enter element (1,2): 4
Enter element (2,0): 3
Enter element (2,1): 2
Enter element (2,2): 1


Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit
```



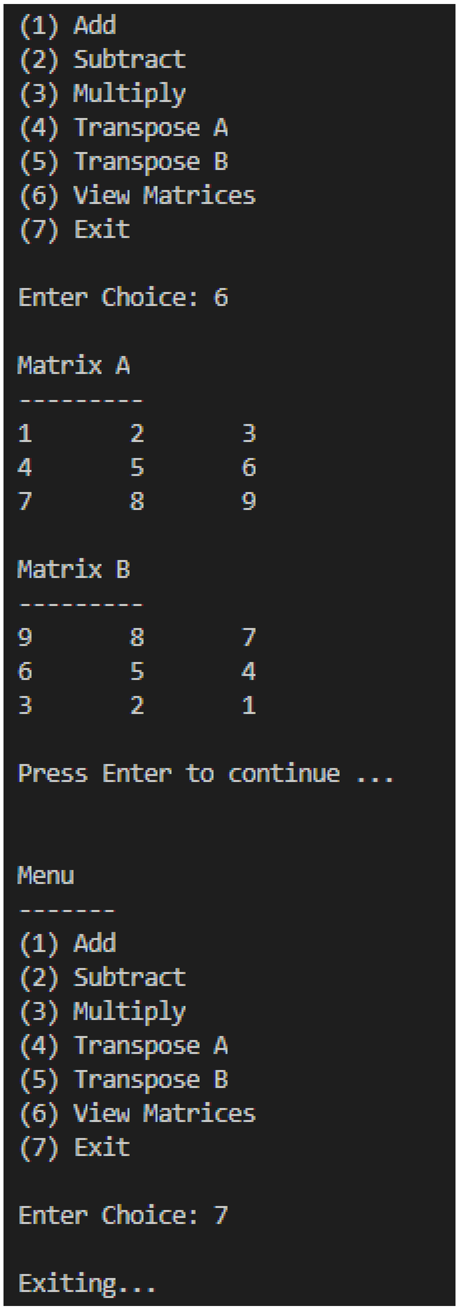
```
Enter Choice: 1

SUM:

10      10      10
10      10      10
10      10      10

Press Enter to continue ...


Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit

Enter Choice: 2

DIFFERENCE:

-8      -6      -4
-2      0       2
4       6       8

Press Enter to continue ...
```



```
Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit

Enter Choice: 3

PRODUCT:

30      24      18
84      69      54
138     114     90

Press Enter to continue ...


Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit
```



```
Enter Choice: 4

TRANSPOSE:

1       4       7
2       5       8
3       6       9

Press Enter to continue ...


Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit

Enter Choice: 5

TRANSPOSE:

9       6       3
8       5       2
7       4       1

Press Enter to continue ...
```



```
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit

Enter Choice: 6

Matrix A
---------
1       2       3
4       5       6
7       8       9

Matrix B
---------
9       8       7
6       5       4
3       2       1

Press Enter to continue ...


Menu
-------
(1) Add
(2) Subtract
(3) Multiply
(4) Transpose A
(5) Transpose B
(6) View Matrices
(7) Exit

Enter Choice: 7

Exiting...
```

Practical - 16

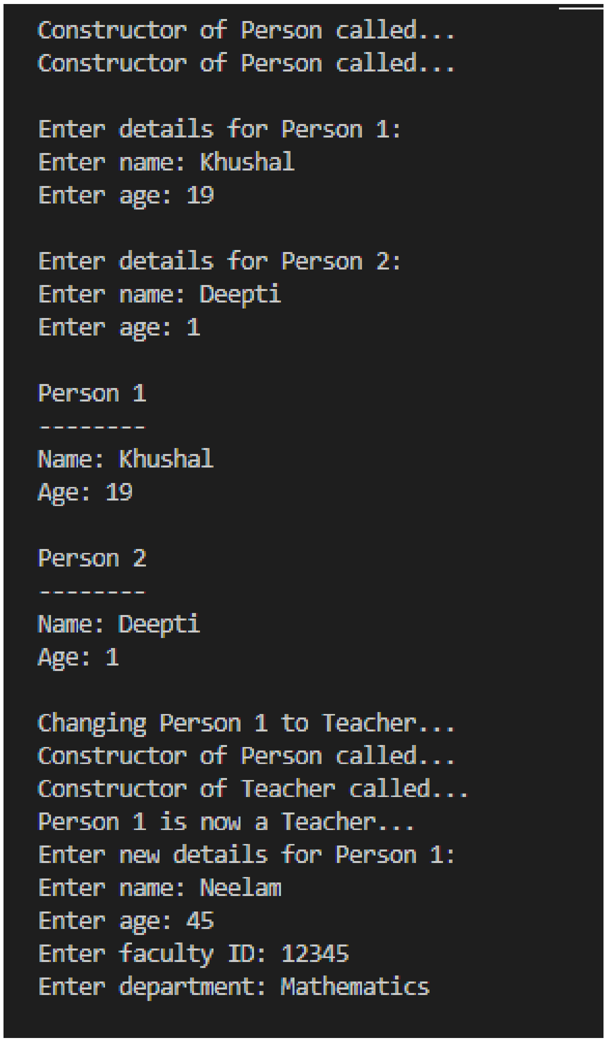
Objective : Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain respective information in the classes and create, display and delete objects of these two classes. (Use Run Time Polymorphism)

Code :

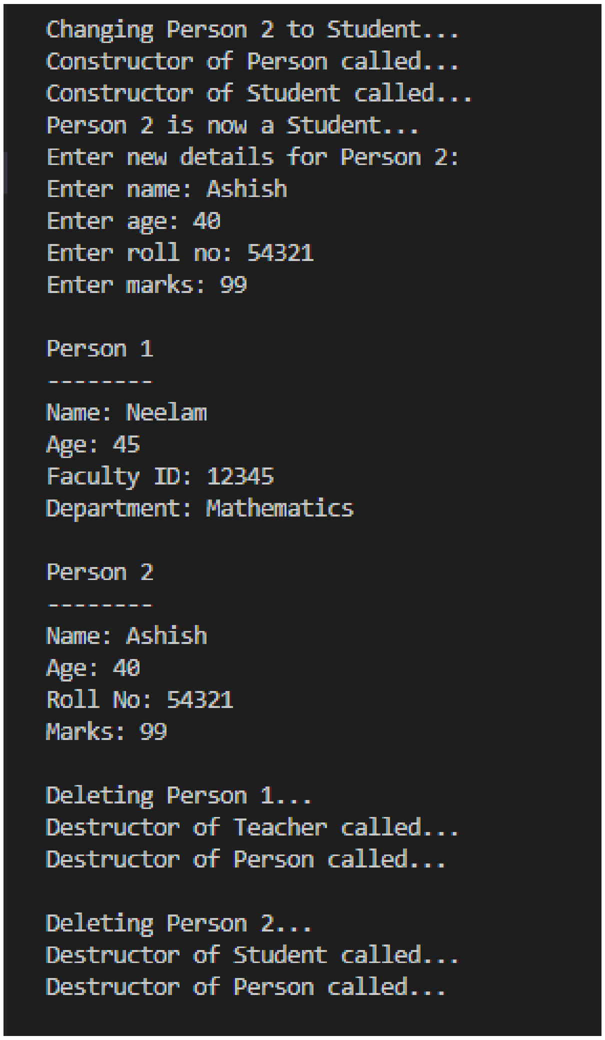
```
/*
QUESTION-16: Create the Person class. Create some objects of this class (by taking
information from the user). Inherit the class Person to create two classes Teacher and
Student class. Maintain respective information in the classes and create, display and
delete objects of these two classes. (Use Run Time Polymorphism)
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <cstring>
using namespace std;
class Person
{
private:
    int age;
    char name[255];

public:
    Person();
    virtual ~Person();
    virtual void get();
    virtual void put();
};
Person::Person()
{
    cout << "Constructor of Person called...\n";
    strcpy(name, "");
    age = 0;
}
```

```
Person::~Person() { cout << "Destructor of Person called...\n"; }
void Person::get()
{
    cout << "Enter name: ";
    cin >> name;
    cout << "Enter age: ";
    cin >> age;
    return;
}
void Person::put()
{
    cout << "Name: " << name << "\n";
    cout << "Age: " << age << "\n";
    return;
}
class Teacher : public Person
{
private:
    int facultyId;
    char department[255];

public:
    Teacher();
    ~Teacher();
    void get();
    void put();
};
Teacher::Teacher()
{
    cout << "Constructor of Teacher called...\n";
    facultyId = 0;
    strcpy(department, "");
}
Teacher::~Teacher() { cout << "Destructor of Teacher called...\n"; }
void Teacher::get()
{
    Person::get();
    cout << "Enter faculty ID: ";
    cin >> facultyId;
    cout << "Enter department: ";
    cin >> department;
    return;
}
void Teacher::put()
{
    Person::put();
    cout << "Faculty ID: " << facultyId << "\n";
    cout << "Department: " << department << "\n";
    return;
}
class Student : public Person
{
private:
    int rollNo;
    float marks;

public:
    Student();
    ~Student();
    void get();
    void put();
};
Student::Student()
{
    cout << "Constructor of Student called...\n";
    rollNo = 0;
    marks = 0;
}
Student::~Student()
{
    cout << "Destructor of Student called...\n";
}
void Student::get()
{
    Person::get();
    cout << "Enter roll no: ";
    cin >> rollNo;
    cout << "Enter marks: ";
    cin >> marks;
    return;
}
void Student::put()
{
    Person::put();
    cout << "Roll No: " << rollNo << "\n";
    cout << "Marks: " << marks << "\n";
    return;
}
int main()
{
    Person *person1, *person2;
    cout << "Initializing two instances of Person...\n";
    person1 = new Person();
    person2 = new Person();
    cout << endl;
    cout << "Enter details for Person 1: \n";
    person1->get();
    cout << endl;
    cout << "Enter details for Person 2: \n";
    person2->get();
    cout << endl;
    cout << "Person 1\n--------\n";
    person1->put();
    cout << endl;
    cout << "Person 2\n--------\n";
    person2->put();
    cout << endl;
    cout << "Changing Person 1 to Teacher...\n";
    person1 = new Teacher();
    cout << "Person 1 is now a Teacher...\n";
    cout << "Enter new details for Person 1:\n";
    person1->get();
    cout << endl;
    cout << "Changing Person 2 to Student...\n";
    person2 = new Student();
    cout << "Person 2 is now a Student...\n";
    cout << "Enter new details for Person 2:\n";
    person2->get();
    cout << endl;
    cout << "Person 1\n--------\n";
    person1->put();
    cout << endl;
    cout << "Person 2\n--------\n";
    person2->put();
    cout << endl;
    cout << "Deleting Person 1...\n";
    delete person1;
    cout << endl;
    cout << "Deleting Person 2...\n";
    delete person2;
    cout << endl;
    return 0;
}
```

Output :



```
Constructor of Person called...
Constructor of Person called...

Enter details for Person 1:
Enter name: Khushal
Enter age: 19

Enter details for Person 2:
Enter name: Deepti
Enter age: 1

Person 1
--------
Name: Khushal
Age: 19

Person 2
--------
Name: Deepti
Age: 1

Changing Person 1 to Teacher...
Constructor of Person called...
Constructor of Teacher called...
Person 1 is now a Teacher...
Enter new details for Person 1:
Enter name: Neelam
Enter age: 45
Enter faculty ID: 12345
Enter department: Mathematics
```



```
Changing Person 2 to Student...
Constructor of Person called...
Constructor of Student called...
Person 2 is now a Student...
Enter new details for Person 2:
Enter name: Ashish
Enter age: 40
Enter roll no: 54321
Enter marks: 99

Person 1
--------
Name: Neelam
Age: 45
Faculty ID: 12345
Department: Mathematics

Person 2
--------
Name: Ashish
Age: 40
Roll No: 54321
Marks: 99

Deleting Person 1...
Destructor of Teacher called...
Destructor of Person called...

Deleting Person 2...
Destructor of Student called...
Destructor of Person called...
```

Practical - 17

Objective : Create a class Triangle. Include overloaded functions for calculating areas. Overload assignment operator and equality operator.

Code :

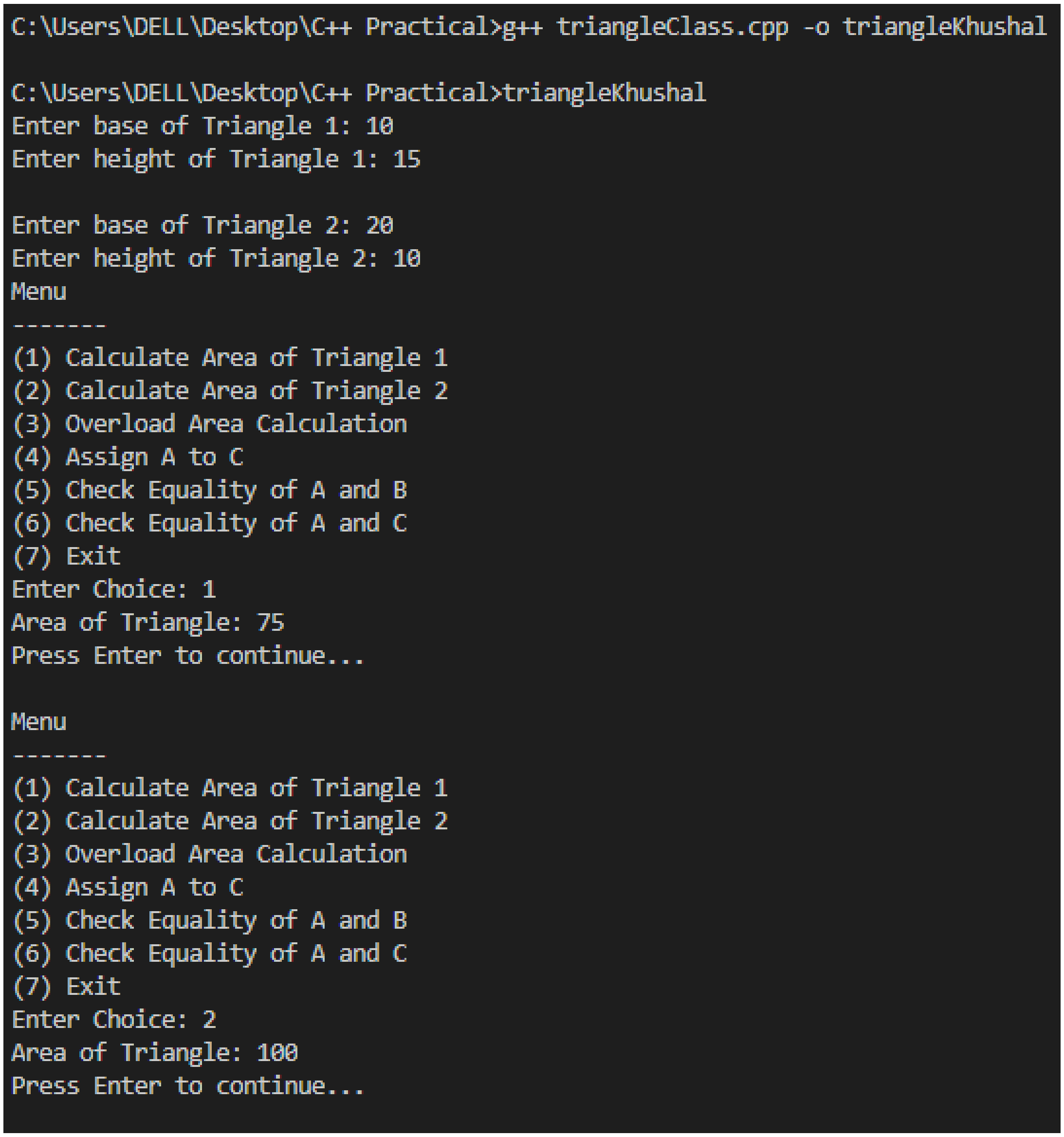
```
/*
QUESTION-17: Create a class Triangle. Include overloaded functions for calculating
areas. Overload assignment operator and equality operator.
Written By: Khushal Sachdeva
*/
#include <iostream>
using namespace std;
class Triangle
{
private:
    int base;
    int height;

public:
    Triangle(int, int);
    void calculateArea();
    void calculateArea(float, float);
    void operator=(Triangle &);
    void operator==(Triangle &);
};
Triangle::Triangle(int x = 0, int y = 0)
{
    base = x;
    height = y;
}
void Triangle::calculateArea()
{
    cout << "Area of Triangle: " << 0.5 * base * height << endl;
    return;
}
void Triangle::calculateArea(float x, float y)
```

```
{
    cout << "Area (Overloaded): " << 0.5 * x * y << endl;
    return;
}
void Triangle::operator=(Triangle &o)
{
    base = o.base;
    height = o.height;
    return;
}
void Triangle::operator==(Triangle &o)
{
    if (base == o.base && height == o.height)
        cout << "Triangles are Equal." << endl;
    else
        cout << "Triangles are Not Equal." << endl;
    return;
}
int main()
{
    int b, h, ch;
    Triangle C;
    cout << "Enter base of Triangle 1: ";
    cin >> b;
    cout << "Enter height of Triangle 1: ";
    cin >> h;
    Triangle A(b, h);
    cout << endl;
    cout << "Enter base of Triangle 2: ";
    cin >> b;
    cout << "Enter height of Triangle 2: ";
    cin >> h;
    Triangle B(b, h);
    do
    {
        cout << "Menu\n-------";
        cout << "\n(1) Calculate Area of Triangle 1"
```

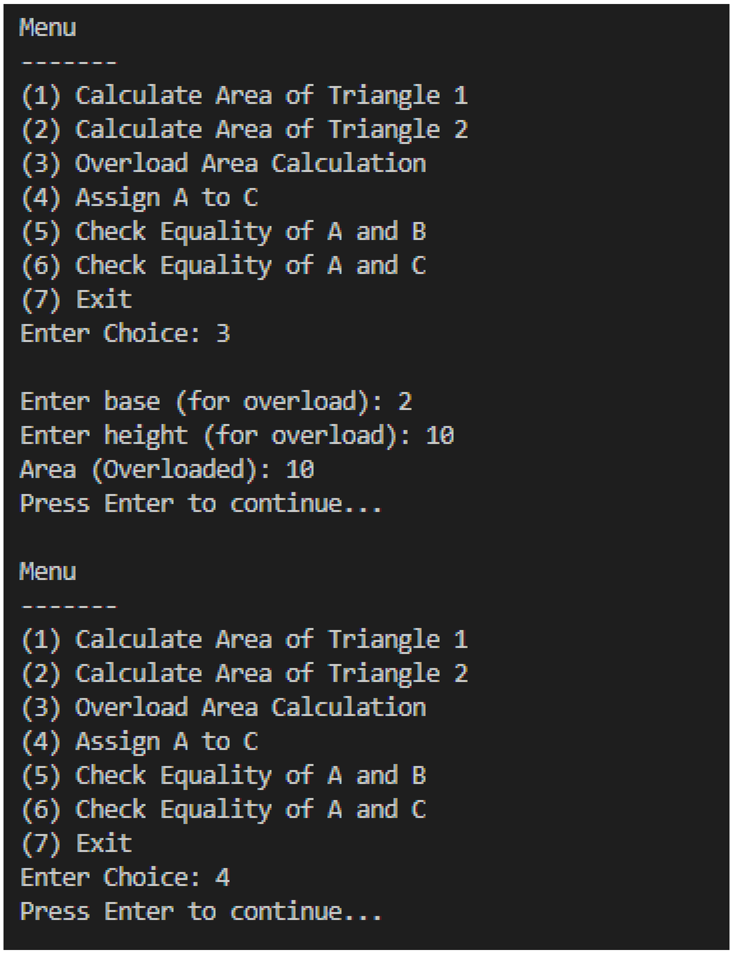
```
     << "\n(2) Calculate Area of Triangle 2"
     << "\n(3) Overload Area Calculation"
     << "\n(4) Assign A to C"
     << "\n(5) Check Equality of A and B"
     << "\n(6) Check Equality of A and C"
     << "\n(7) Exit\n"
     << "Enter Choice: ";
cin >> ch;
switch (ch)
{
case 1:
    A.calculateArea();
    break;
case 2:
    B.calculateArea();
    break;
case 3:
    cout << "\nEnter base (for overload): ";
    cin >> b;
    cout << "Enter height (for overload): ";
    cin >> h;
    A.calculateArea(b, h);
    break;
case 4:
    C = A;
    break;
case 5:
    A == B;
    break;
case 6:
    A == C;
    break;
case 7:
default:
    break;
}
if (ch != 7)
```

```
        {
            cout << "Press Enter to continue...\n";
            cin.ignore();
            cin.get();
        }
    } while (ch != 7);
    cout << "\nExiting...\n";
    return 0;
}
```

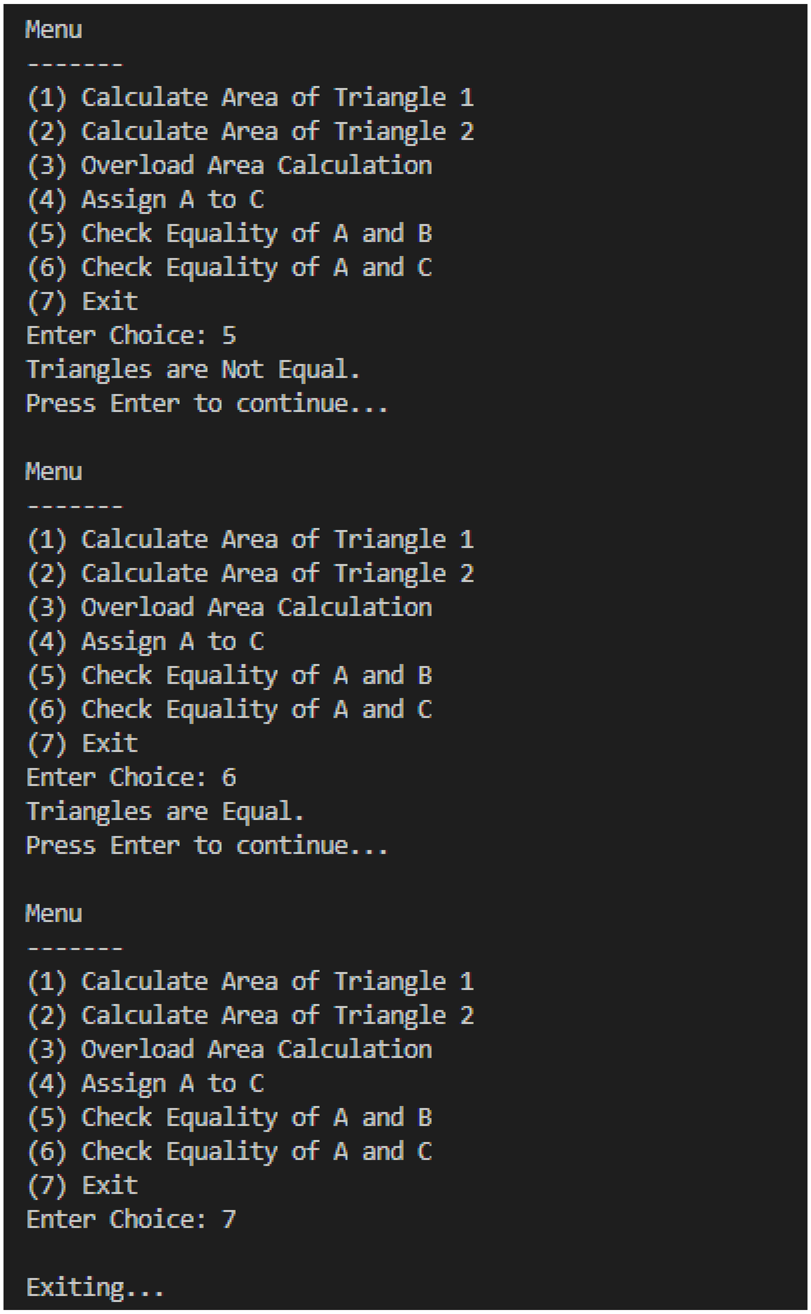
Output :



```
C:\Users\DELL\Desktop\C++ Practical>g++ triangleClass.cpp -o triangleKhushal

C:\Users\DELL\Desktop\C++ Practical>triangleKhushal
Enter base of Triangle 1: 10
Enter height of Triangle 1: 15

Enter base of Triangle 2: 20
Enter height of Triangle 2: 10
Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 1
Area of Triangle: 75
Press Enter to continue...

Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 2
Area of Triangle: 100
Press Enter to continue...
```



```
Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 3

Enter base (for overload): 2
Enter height (for overload): 10
Area (Overloaded): 10
Press Enter to continue...

Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 4
Press Enter to continue...
```



```
Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 5
Triangles are Not Equal.
Press Enter to continue...

Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 6
Triangles are Equal.
Press Enter to continue...

Menu
-------
(1) Calculate Area of Triangle 1
(2) Calculate Area of Triangle 2
(3) Overload Area Calculation
(4) Assign A to C
(5) Check Equality of A and B
(6) Check Equality of A and C
(7) Exit
Enter Choice: 7

Exiting...
```

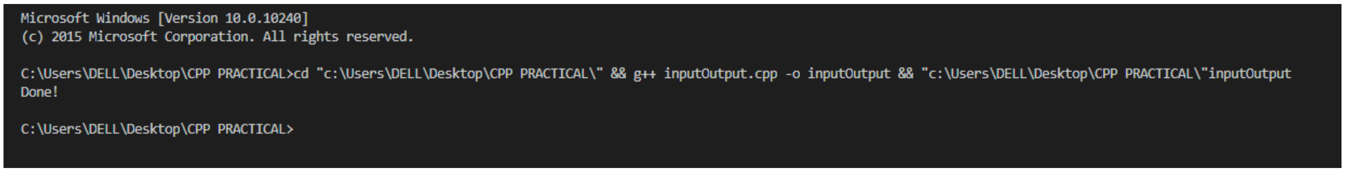
Practical - 18

Objective : Copy the contents of one text file to another file, after removing all whitespaces.

Code :

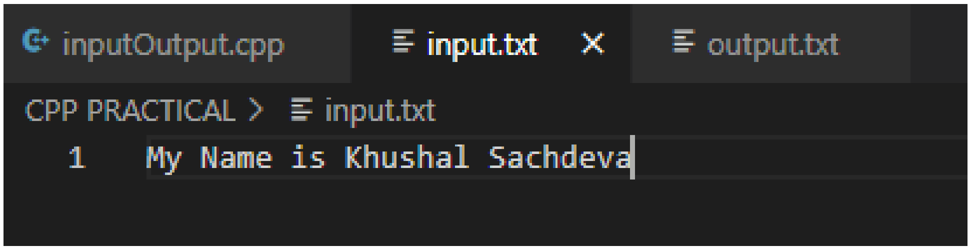
```
/*
QUESTION-18: Copy the contents of one text file to another file, after removing all
whitespaces
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <fstream>
using namespace std;
main()
{
    char arr;
    ifstream obj("input.txt");
    if (obj.fail())
    {
        cerr << "Input file does not exist!!" << endl;
        return 0;
    }
    ofstream obj2("output.txt");
    if (obj2.fail())
    {
        cerr << "Output file cannot be created!!" << endl;
        return 0;
    }
    obj.get(arr);
    do
    {
        if (arr != ' ')
        {
            obj2 << arr;
        }
    } while (obj.get(arr));
    obj.close();
    obj2.close();
    cout << "Done!" << endl;
    return 0;
}
```

Output :



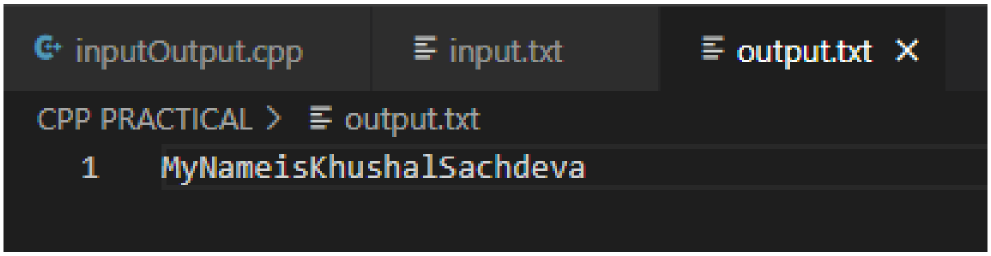
```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\CPP PRACTICAL>cd "c:\Users\DELL\Desktop\CPP PRACTICAL\" && g++ inputOutput.cpp -o inputOutput && "c:\Users\DELL\Desktop\CPP PRACTICAL\"inputOutput
Done!

C:\Users\DELL\Desktop\CPP PRACTICAL>
```

Input.txt :



```
My Name is Khushal Sachdeva
```

Output.txt :



```
MyNameisKhushalSachdeva
```

Practical - 19

Objective : Create a class Box containing length, breadth and height. Include the following methods in it:

(a) Calculate Surface Area

(b) Calculate Volume

(c) Add, Overload + operator (to add two boxes length, breadth and height)

(d) Overload operator = = (to check equality of two boxes), as a friend function

(e) Overload Assignment operator

(f) Check if it is a cube or cuboid

Write a program which takes input from the user for length, breadth and height to test the above class.

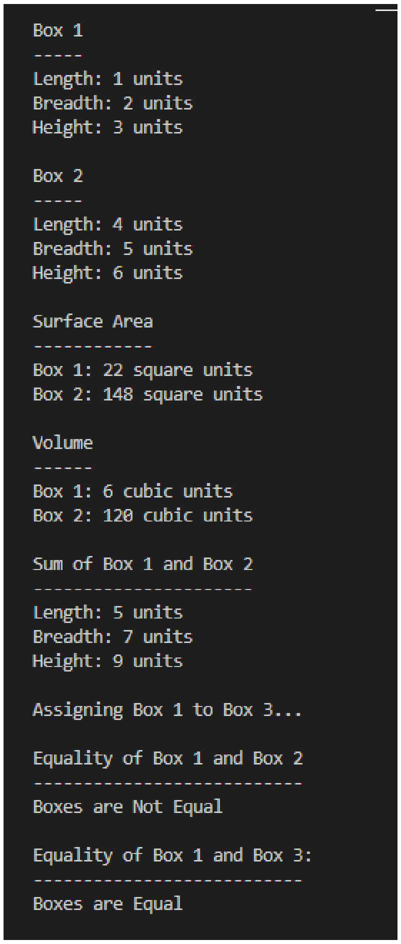
Code :

```
/*
QUESTION-19: Create a class Box containing length, breadth and height. Include the
following methods in it:
(a) Calculate Surface Area
(b) Calculate Volume
(c) Add, Overload + operator (to add two boxes length, breadth and height)
(d) Overload operator = = (to check equality of two boxes), as a friend function
(e) Overload Assignment operator
(f) Check if it is a cube or cuboid
Write a program which takes input from the user for length, breadth and height to test
the above class.
Written By: Khushal Sachdeva
*/

#include <iostream>
using namespace std;
class Box
{
private:
    float length;
    float breadth;
    float height;

public:
    Box();
    Box(Box &);
    Box(float, float, float);
    ~Box() {}
    void show();
    void isCube();
    float calculateVolume();
    float calculateSurfaceArea();
    Box operator+(Box &);
    Box operator=(Box &);
    friend void operator==(Box &, Box &);
};
Box::Box()
{
    length = breadth = height = 0;
}
Box::Box(Box &o)
{
    length = o.length;
    breadth = o.breadth;
    height = o.height;
}
Box::Box(float a, float b, float c)
{
    length = a;
    breadth = b;
    height = c;
}
void Box::show()
{
    cout << "Length: " << length << " units\n";
    cout << "Breadth: " << breadth << " units\n";
    cout << "Height: " << height << " units\n";
    return;
```

```
}
void Box::isCube()
{
    if (length == breadth && breadth == height && height == length)
        cout << "Box is a Cube";
    else
        cout << "Box is a Cuboid";
    cout << endl;
    return;
}
float Box::calculateVolume() { return length * breadth * height; }
float Box::calculateSurfaceArea() { return 2 * (length * breadth + breadth * height +
height * length); }
Box Box::operator+(Box &o)
{
    Box temp(*this);
    temp.length += o.length;
    temp.breadth += o.breadth;
    temp.height += o.height;
    return temp;
}
Box Box::operator=(Box &o)
{
    length = o.length;
    breadth = o.breadth;
    height = o.height;
    return *this;
}
void operator==(Box &a, Box &o)
{
    if (a.length == o.length && a.breadth == o.breadth && a.height == o.height)
        cout << "Boxes are Equal";
    else
        cout << "Boxes are Not Equal";
    cout << endl;
    return;
}
```

```
int main()
{
    Box C;
    int l, b, h;
    cout << "Enter dimensions of Box 1: ";
    cin >> l >> b >> h;
    Box A(l, b, h);
    cout << "Enter dimensions of Box 2: ";
    cin >> l >> b >> h;
    Box B(l, b, h);
    cout << endl;
    cout << "Box 1\n";
    cout << "-----\n";
    A.show();
    cout << endl;
    cout << "Box 2\n";
    cout << "-----\n";
    B.show();
    cout << endl;
    cout << "Surface Area\n";
    cout << "------------\n";
    cout << "Box 1: " << A.calculateSurfaceArea() << " square units" << endl;
    cout << "Box 2: " << B.calculateSurfaceArea() << " square units" << endl;
    cout << endl;
    cout << "Volume\n";
    cout << "------\n";
    cout << "Box 1: " << A.calculateVolume() << " cubic units" << endl;
    cout << "Box 2: " << B.calculateVolume() << " cubic units" << endl;
    cout << endl;
    cout << "Sum of Box 1 and Box 2\n";
    cout << "----------------------\n";
    (A + B).show();
    cout << endl;
    cout << "Assigning Box 1 to Box 3...\n\n";
    C = A;
    cout << "Equality of Box 1 and Box 2\n";
    cout << "---------------------------\n";
    A == B;
    cout << endl;
    cout << "Equality of Box 1 and Box 3:\n";
    cout << "---------------------------\n";
    A == C;
    cout << endl;
    return 0;
}
```

Output :



```
Box 1
-----
Length: 1 units
Breadth: 2 units
Height: 3 units

Box 2
-----
Length: 4 units
Breadth: 5 units
Height: 6 units

Surface Area
------------
Box 1: 22 square units
Box 2: 148 square units

Volume
------
Box 1: 6 cubic units
Box 2: 120 cubic units

Sum of Box 1 and Box 2
----------------------
Length: 5 units
Breadth: 7 units
Height: 9 units

Assigning Box 1 to Box 3...

Equality of Box 1 and Box 2
---------------------------
Boxes are Not Equal

Equality of Box 1 and Box 3:
---------------------------
Boxes are Equal
```

Practical - 20

Objective : Create a class Length containing feet and inches. Include following functions in it:

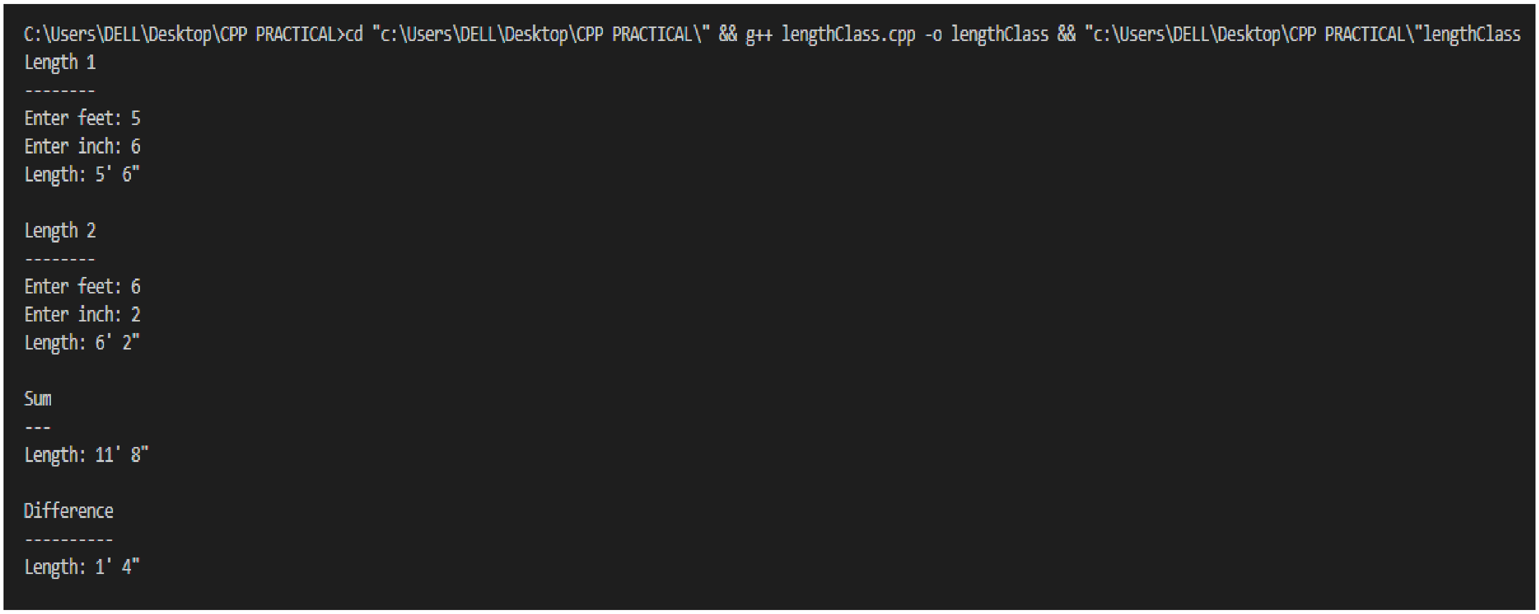
(a) Input Length object
(b) Overload + operator (to add two lengths)
(c) Overload - operator (to subtract two lengths)
(d) Display Length object

Write a program which takes input from the user for feet and inches to test the above class.

Code :

```
/*
QUESTION-20: Create a class Length containing feet and inch. Include following
functions in it:
(a) Input Length object
(b) Overload + operator (to add two lengths)
(c) Overload - operator (to subtract two lengths)
(d) Display Length object
Write a program which takes input from user for feet and inch to test the above class.
Written By: Khushal Sachdeva
*/
#include <iostream>
#include <cstdlib>
using namespace std;
class Length
{
private:
    int feet;
    int inch;

public:
    Length();
    Length(Length &);
    ~Length() {}
    void input();
    void display();
    Length operator+(Length &);
    Length operator-(Length &);
};
Length::Length() { feet = inch = 0; }
Length::Length(Length &o)
{
    feet = o.feet;
    inch = o.inch;
}
void Length::input()
{
    cout << "Enter feet: ";
    cin >> feet;
    cout << "Enter inch: ";
    cin >> inch;
    if (inch >= 12)
        while (inch >= 12)
        {
            inch -= 12;
            feet++;
        }
    return;
}
void Length::display()
{
    cout << "Length: ";
    cout << feet << "\' ";
    cout << inch << "\"\n";
    return;
}
Length Length::operator+(Length &o)
{
    Length temp(*this);
    temp.feet += o.feet;
    temp.inch += o.inch;
    if (temp.inch >= 12)
        while (temp.inch >= 12)
        {
            temp.inch -= 12;
            temp.feet++;
        }
    return temp;
}
Length Length::operator-(Length &o)
{
    Length temp(*this);
    temp.feet -= o.feet;
    temp.inch -= o.inch;
    if (temp.inch >= 12)
        while (temp.inch >= 12)
        {
            temp.inch -= 12;
            temp.feet++;
        }
    temp.feet = abs(temp.feet);
    temp.inch = abs(temp.inch);
    return temp;
}
int main()
{
    Length A, B, sum, diff;
    cout << "Length 1\n";
    cout << "---------\n";
    A.input();
    A.display();
    cout << endl;
    cout << "Length 2\n";
    cout << "---------\n";
    B.input();
    B.display();
    cout << endl;
    cout << "Sum\n";
    cout << "---\n";
    sum = A + B;
    sum.display();
    cout << endl;
    cout << "Difference\n";
    cout << "----------\n";
    diff = A - B;
    diff.display();
    cout << endl;
    return 0;
}
```

Output :



```
C:\Users\DELL\Desktop\CPP PRACTICAL>cd "c:\Users\DELL\Desktop\CPP PRACTICAL\" && g++ lengthClass.cpp -o lengthClass && "c:\Users\DELL\Desktop\CPP PRACTICAL\"lengthClass
Length 1
--------
Enter feet: 5
Enter inch: 6
Length: 5' 6"

Length 2
--------
Enter feet: 6
Enter inch: 2
Length: 6' 2"

Sum
---
Length: 11' 8"

Difference
----------
Length: 1' 4"
```