

NATIONAL UNIVERSITY OF TECHNOLOGY, ISLAMABAD



Programing Fundamentals (CS121)

LAB: 14

Submitted to : Ma'am Kainat Babar

Submitted by : Fariah Hajra

NUTECH ID : F24605037

Session : Fall 24

Class : BSCS 24-A

Submission Date : November 11th,2024

FILE HANDLING AND RECURSIVE FUNCTIONS

TASK 1:

Perform the following operations on a named data.text:

- Create the file and write a message into it.
- Read and display the content of the file.
- Append a new message to the file.
- Count the number of words in the file.
- Rename the file to info.txt.
- Delete the file.

CODE:

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

    //create file, write
    ofstream obj("C:\\Users\\Student\\Desktop\\file.txt");
    obj<< "this is my first file \n";
    cout<<"Written successfully"<<endl;
    obj.close();

    //read from file

    ifstream obj1("C:\\Users\\Student\\Desktop\\file.txt");
    if(obj1.is_open())
    {

        string st;
        while(getline(obj1,st))
        {
            cout<<st<<" ";
        }
        obj1.close();
    }
    else
    {
        cout<<"unable to open the file \n";
        return 1;
    }

    //Append data into file
    ofstream appx("C:\\Users\\Student\\Desktop\\file.txt", ios::app);
    appx<<"this is appended line \n";
    cout<<"\n Appended successfully";
    appx.close();
}
```

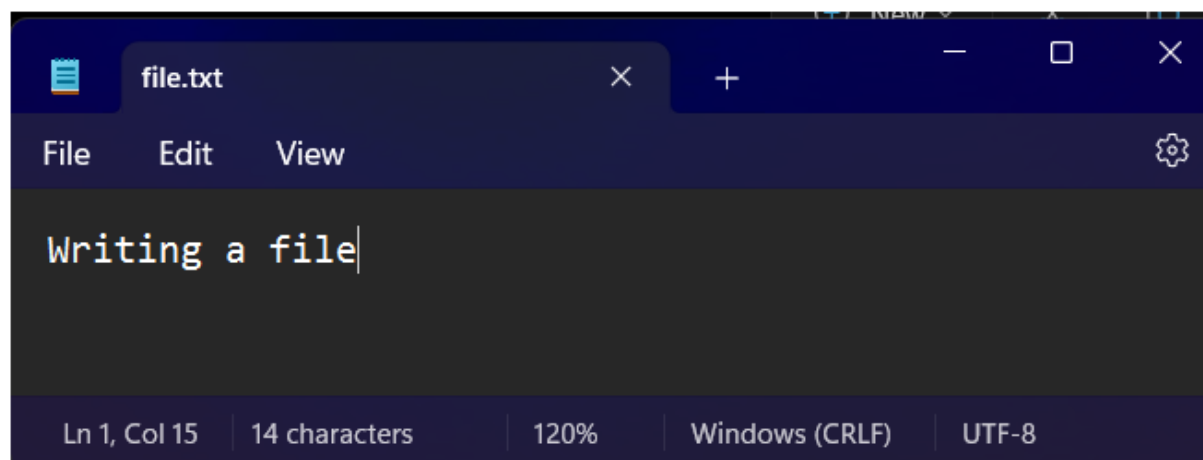
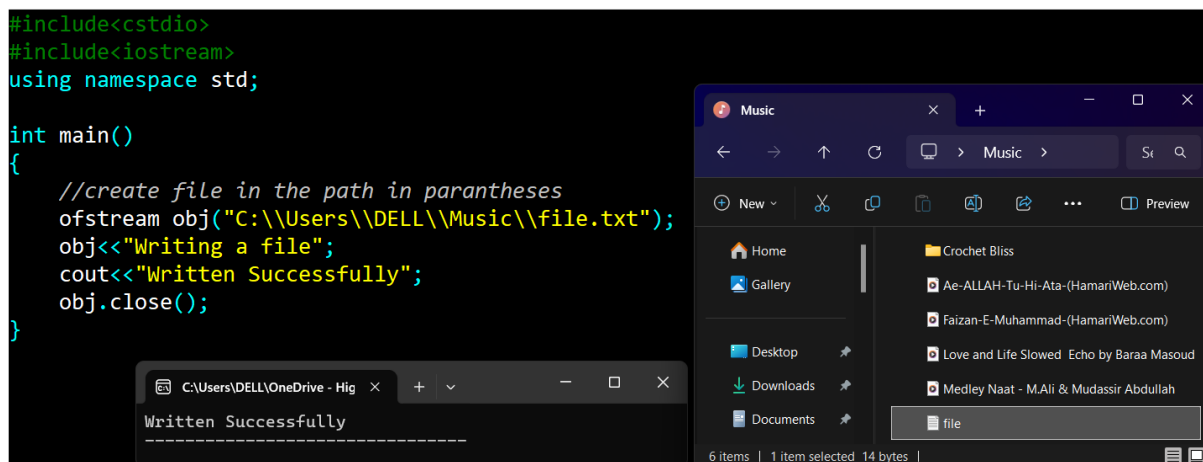
```

//Rename file
string oldfile= "C:\\Users\\Student\\Desktop\\file.txt";
string newfile = "C:\\Users\\Student\\Desktop\\newfile.txt";
rename(oldfile.c_str(), newfile.c_str());
cout<<"\n renamed successfully";

//delete file
//string filename= "C:\\Users\\Student\\Desktop\\file.txt";
if(remove(filename.c_str())==0)
{
    cout<<"file deleted";
}
else
{
    cout<<"unable to open the file";
    return 1;
}

```

OUTPUT:



TASK 2:

Write a recursive function to calculate the factorial of a given number n. Test the function with different values of n.

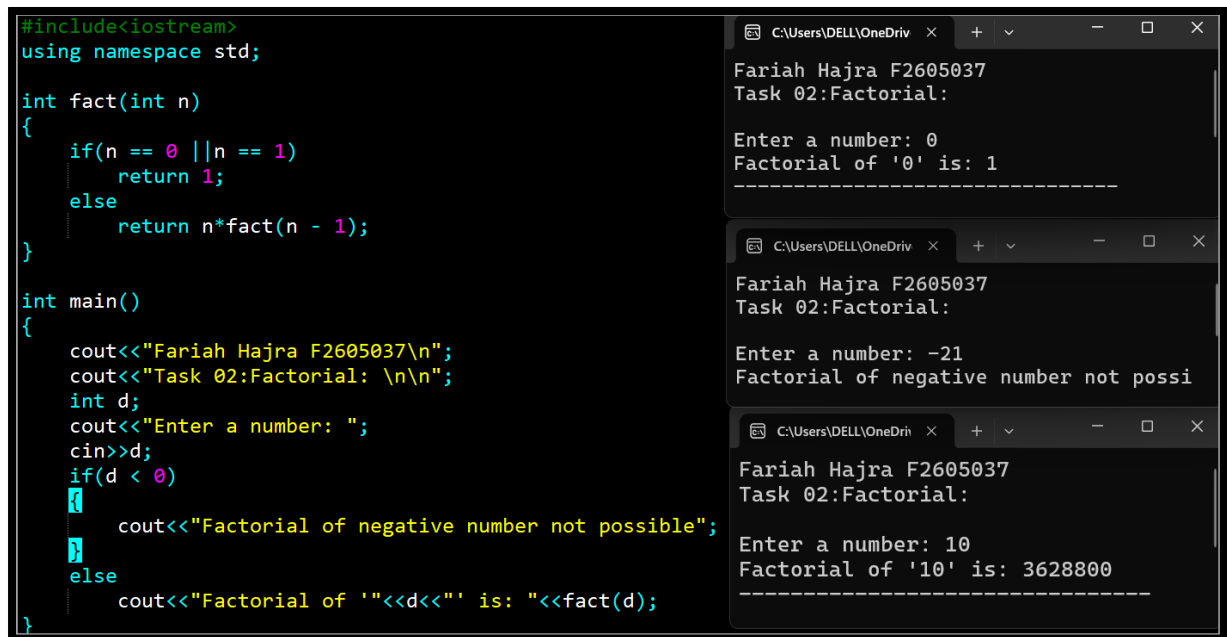
CODE:

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

int fact(int n)
{
    if(n == 0 || n == 1)
        return 1;
    else
        return n*fact(n - 1);
}

int main()
{
    cout<<"Fariah Hajra F2605037\n";
    cout<<"Task 02:Factorial: \n\n";
    int d;
    cout<<"Enter a number: ";
    cin>>d;
    if(d < 0)
    {
        cout<<"Factorial of negative number not possible";
    }
    else
        cout<<"Factorial of '"<<d<<"' is: "<<fact(d);
}
```

OUTPUT:



The image shows a C++ program for calculating factorials and three separate console window screenshots demonstrating its execution with different inputs.

Code Snippet:

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

int fact(int n)
{
    if(n == 0 || n == 1)
        return 1;
    else
        return n*fact(n - 1);
}

int main()
{
    cout<<"Fariah Hajra F2605037\n";
    cout<<"Task 02:Factorial: \n\n";
    int d;
    cout<<"Enter a number: ";
    cin>>d;
    if(d < 0)
    {
        cout<<"Factorial of negative number not possible";
    }
    else
        cout<<"Factorial of '"<<d<<"' is: "<<fact(d);
}
```

Test Case 1:

```
C:\Users\DELL\OneDrive x + - □ x
Fariah Hajra F2605037
Task 02:Factorial:

Enter a number: 0
Factorial of '0' is: 1
-----
```

Test Case 2:

```
C:\Users\DELL\OneDrive x + - □ x
Fariah Hajra F2605037
Task 02:Factorial:

Enter a number: -21
Factorial of negative number not possi
```

Test Case 3:

```
C:\Users\DELL\OneDrive x + - □ x
Fariah Hajra F2605037
Task 02:Factorial:

Enter a number: 10
Factorial of '10' is: 3628800
-----
```

TASK 3:

Write a recursive function to generate the Fibonacci series up to the n-th term. Use this function in a program to display the series for a user-specified value of n.

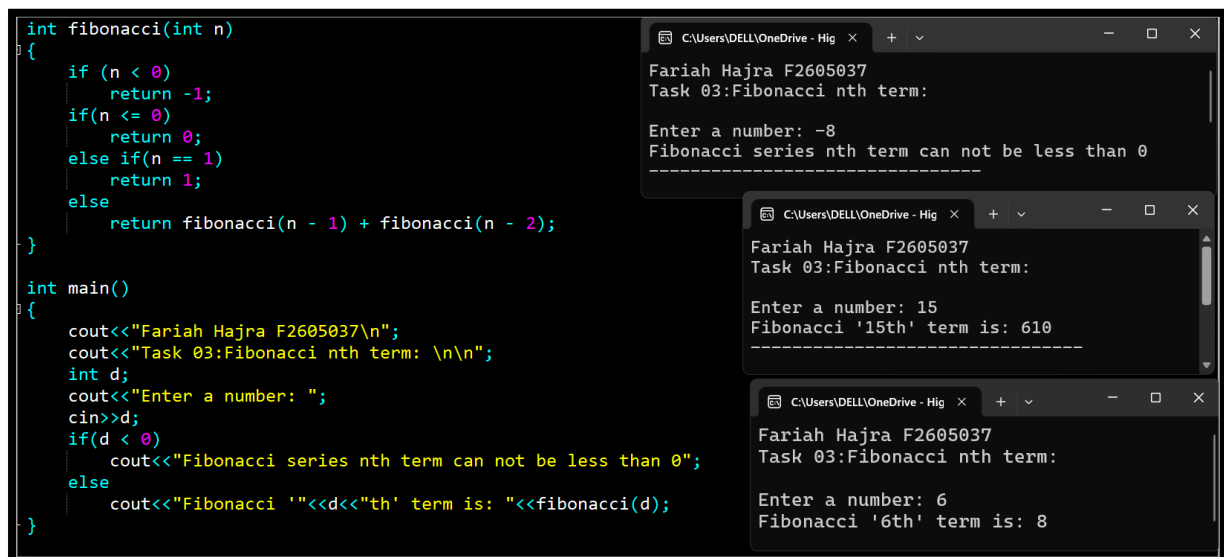
CODE:

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

int fibonacci(int n)
{
    if (n < 0)
        return -1;
    if(n <= 0)
        return 0;
    else if(n == 1)
        return 1;
    else
        return fibonacci(n - 1) + fibonacci(n - 2);
}

int main()
{
    cout<<"Fariah Hajra F2605037\n";
    cout<<"Task 03:Fibonacci nth term: \n\n";
    int d;
    cout<<"Enter a number: ";
    cin>>d;
    if(d < 0)
        cout<<"Fibonacci series nth term can not be less than 0";
    else
        cout<<"Fibonacci '"<<d<<"th' term is: "<<fibonacci(d);
}
```

OUTPUT:



The screenshot displays the C++ source code for a recursive Fibonacci function and its main function, alongside three separate console window captures showing the program's execution. The code defines a recursive function `fibonacci` that returns -1 for negative inputs, 0 for zero, 1 for one, and the sum of the two preceding terms for larger values. The `main` function prints the user's name and task number, prompts for an integer `d`, and then either displays an error message for negative `d` or the Fibonacci value for `d`.

The first console window shows the program running with input -8, resulting in the error message: "Fibonacci series nth term can not be less than 0". The second window shows input 15, resulting in the output: "Fibonacci '15th' term is: 610". The third window shows input 6, resulting in the output: "Fibonacci '6th' term is: 8".

TASK 4:

Define a structure named Student with the following attributes: Name, Roll Number, and Marks.

- **Create an array of Student structures to store details of 5 students.**
- **Write a program to input details for these 5 students from the user.**
- **Display the details of all students who scored more than 75 marks.**

CODE:

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

struct scholar
{
    string name;
    int rollnum;
    int marks;
};

int main()
{
    cout<<"Fariah Hajra F2605037\n";
    cout<<"Task 04:Enetring data using Strucure: \n\n";
    int scholar.data[5];
    cout<<"Enter data of students(name, rollno., marks): \n";
    for(int i = 0;i < 4;i ++)
    {
        cin>>d[i].name;
        cin>>d[i].rollnum;
        cin>>d[i].marks;
    }
    cout<<"Data eneted having marks >75 is:  "
    for(int i = 0;i <= 4;i++)
    {
        if(d[i].marks >= 75)
        {
            cout<<"student"<<d[i].name;
        }
        else
            cout<<"Factorial of '"<<d<<"' is: "<<fact(d);
    }
}
```

OUTPUT: