ASSIGNMENT #2

PAIR MEMBERS:

Ahmed Jamil - 153169

Fahad Ali Khan - 153177

CLASS : BSCS 5th

SUBJECT : Software Engineering

TEACHER : Sir, Ahmad Moshin

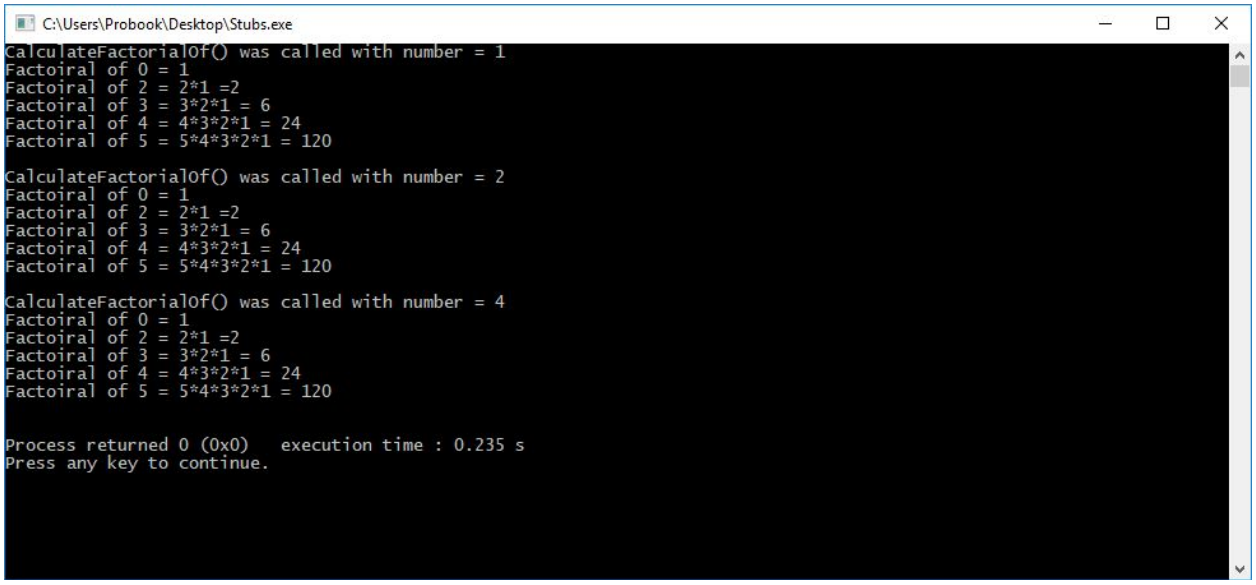
```cpp
#include <iostream>
using namespace std;

//Stub Helpers

void Zero_Factorial_Is_One()
{ cout << "Factoiral of 0 = " << 1 << endl; }

void One_Factorial_Is_One()
{ cout << "Factoiral of 1 = " << 1 << endl; }

void Two_Factorial_Is_Two()
{ cout << "Factoiral of 2 = 2*1 =" << 2*1 << endl; }

void Three_Factorial_Is_Six()
{ cout << "Factoiral of 3 = 3*2*1 = " << 3*2*1 << endl; }

void Four_Factorial_Is_TwentyFour()
{ cout << "Factoiral of 4 = 4*3*2*1 = " << 4*3*2*1 << endl; }

void Five_Factorial_Is_OneTwenty()
{ cout << "Factoiral of 5 = 5*4*3*2*1 = " << 5*4*3*2*1 << endl; }



//Stub
int CalculateFactorialOf(int number)
{
    //Dummy Output
    cout << "CalculateFactorialOf() was called with number = " <<number <<endl;
    Zero_Factorial_Is_One();
    Two_Factorial_Is_Two();
    Three_Factorial_Is_Six();
    Four_Factorial_Is_TwentyFour();
    Five_Factorial_Is_OneTwenty();
    cout << endl;

}


int main()
{
    CalculateFactorialOf(1);
    CalculateFactorialOf(2);
    CalculateFactorialOf(4);

    return 0;
}


```

```cpp
//----------- DRIVER PROGRAM ---------------------//

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


int CalculateFactorialOf(int number)
{
    assert(number >= 0);

    int fact = 1;
    for(int i = 2; i<=number ;i++)
        fact = fact*i;
    return fact;
}


int main()
{
    const int factorial_of_zero = 1;
    const int factorial_of_one = 1;
    const int factorial_of_two = 2;
    const int factorial_of_three = 6;
    const int factorial_of_four = 24;
    const int factorial_of_five = 120;

    int result;

    cout << "Testing CalculateFactorialOf() with number = " << 0 <<endl;
    result = CalculateFactorialOf(0);
    cout << "Factorial = " << result << endl << endl;
    assert(result == factorial_of_zero);

    cout << "Testing CalculateFactorialOf() with number = " << 1 <<endl;
    result = CalculateFactorialOf(1);
    cout << "Factorial = " << result <<endl << endl;
    assert(result == factorial_of_one);

    cout << "Testing CalculateFactorialOf() with number = " << 2 <<endl;
    result = CalculateFactorialOf(2);
    cout << "Factorial = " << result <<endl << endl;
    assert(result == factorial_of_two);

    cout << "Testing CalculateFactorialOf() with number = " << 3 <<endl;
    result = CalculateFactorialOf(3);
    cout << "Factorial = " << result <<endl << endl;
    assert(result == factorial_of_three);

    cout << "Testing CalculateFactorialOf() with number = " << 4 <<endl;
    result = CalculateFactorialOf(4);
    cout << "Factorial = " << result <<endl << endl;
    assert(result == factorial_of_four);

    cout << "Testing CalculateFactorialOf() with number = " << 5 <<endl;
    result = CalculateFactorialOf(6);
    cout << "Factorial = " << result <<endl << endl;
    assert(result == factorial_of_five);

    return 0;
}
```

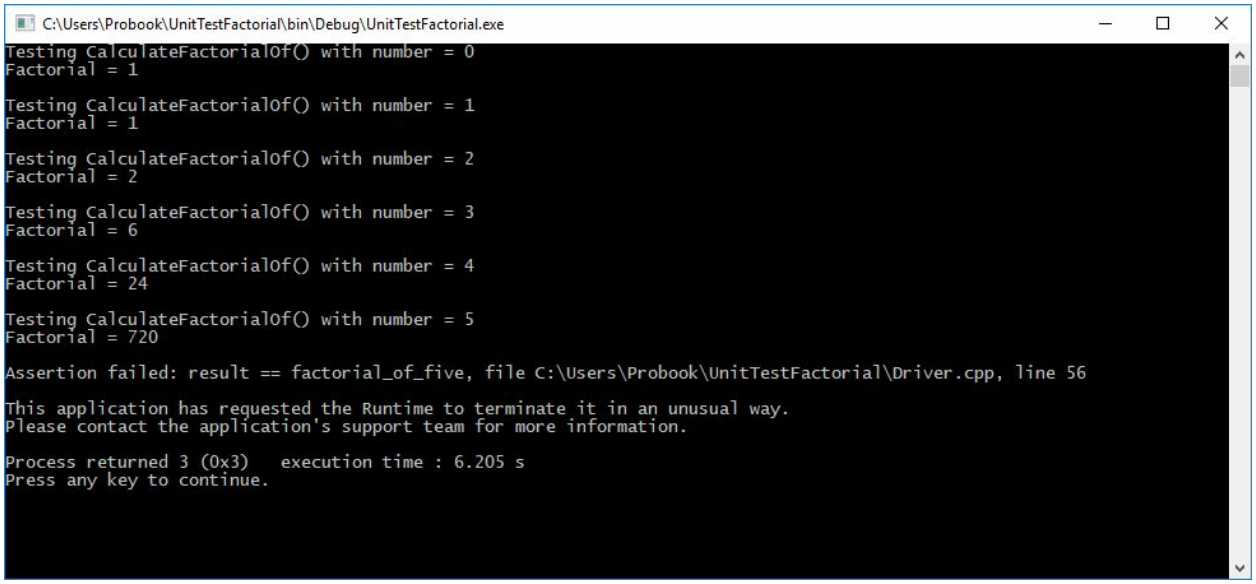
```cpp
// -- FINAL APPLICATION AFTER UNIT TESTING --


#include <iostream>
using namespace std;


int CalculateFactorialOf(int number)
{
    int fact = 1;
    for(int i = 2; i<=number ;i++)
        fact = fact*i;
    return fact;
}


int main()
{
    int number;
    int result;
    cout << "Factoiral Application \n";
    cout << "====================== \n\n";
    cout << "Enter a number : ";
    cin  >> number;

    result = CalculateFactorialOf(number);

    cout << "Factorial = " << result << endl;
    return 0;
}


```

OUTPUTS

Stubs OUTPUT



Drivers with assert usage OUTPUT



Final Application After Unit Testing OUTPUT