CS1050 – Lab 5

Fall 2021

Concepts to Practice

- User-written functions
- math.h

Submission Information

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab using the lab code given by the TA.

Use the following submit command:

```
mucs submit <class> <assignment_type> <filename>
```

For example:

```
mucs submit 1050 lab5 lab5.c
```

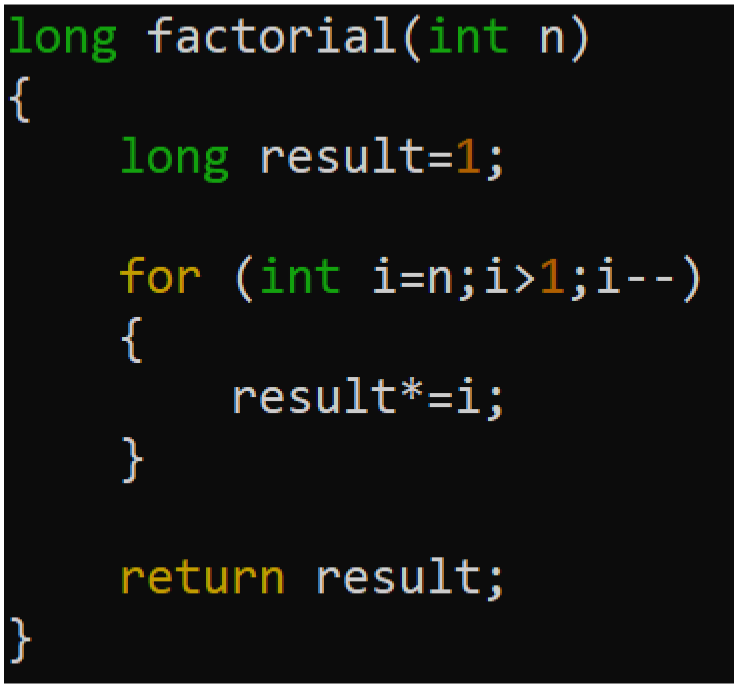
Description

For the lab assignment, you are to write a function that returns the mathematical number e as a double and prints it from main(). Recall that e is the base of the natural logarithm and is approximately 2.7182818285. e is an irrational number, similar to π. You should be familiar with π from calculating the area and circumference of a circle. Your approximation of e should be correct to 10 decimal places.

The formula for approximating e is:

```
e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + . . .
```

Unless you are doing the honors extension, you may not include math.h. Notice the terms on the bottom of each quotient are factorials. You can use this factorial function if you want (it might look familiar from my video):



```
long factorial(int n)
{
    long result=1;

    for (int i=n;i>1;i--)
    {
        result*=i;
    }

    return result;
}
```

Honors Extension

Instead of just being able to print e, your program should be able to print e to any integer power from 1 up through 5. It is OK to use the pow() function from math.h, but no other function from math.h can be used. However, it is OK to compare your results to the exp() function but you can't actually use the exp() function to get results for your function. You need only be accurate to 3 decimal places for the honors extension.

Basic Requirements (you will lose points if you do not have these)

Your program should have a comment header at the top of the file that provides the lab number, the course, the semester, the author, and the date when the program was written. This comment header should be nicely formatted and easy to read.

All code should be indented properly and should be easy to read.

Hints

- You can use "%lf" with printf() to print a double value. You can use "%.10lf" with printf() to print a double value with 10 digits to the right of the decimal. "%.3lf" will print with 3 digits to the right of the decimal.
- For the honors extension, you might want to look up the Taylor series for approximating e^x. I just showed this formula for e^1, but it generalizes for any x. Alternatively, consider how you could just use pow() combined with the non-honors result.

Sample Output (non-honors)

```
$ .a.out
e=2.7182818285
```

Sample Output (honors)

```
$ .a.out
e to the 1 power = 2.718
e to the 2 power = 7.389
e to the 3 power = 20.086
e to the 4 power = 54.598
e to the 5 power = 148.413
```

Guidelines for Grading Lab 5
40 Points Possible (no bonus points)

General

If your program does not compile, or produce any input/output then your lab will receive a grade of **zero points**. You will receive **zero points** if your program merely simulates functioning correctly (for example, if you just use printf to make the output match a sample run, you get **zero**). Note that you must have a function that does the actual work of approximating e (you cannot just print out the literal digits we have discussed), and you may NOT do the work of approximating e in your main() function.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 5"). Your code should be nicely indented. You should include comments throughout your program that explain what you are intending to do. **You will lose up to 10 points if you do not meet these basic requirements**.

Non-honors

10 points: The main() function calls a function to approximate e and prints whatever this function returns..
10 points: The function simulating e returns a close approximation of e.
10 points: The function simulating e is correct to 5 digits.
10 points: The function simulating e is correct to 10 digits.

Honors

10 points: The main() function calls a function to approximate e and prints whatever this function returns..
10 points: The function simulating e is correct to 3 digits for e^1
5 points: The function simulating e is correct to 3 digits for e^2
5 points: The function simulating e is correct to 3 digits for e^3
5 points: The function simulating e is correct to 3 digits for e^4
5 points: The function simulating e is correct to 3 digits for e^5