COP 3014 Honors: Spring 2017
Homework 4

Total Points: 100 (plus 50 point extra credit)
Due: Thursday 02/23/2017 11:59:59 PM

1 Objective

The purpose of this assignment is to test your familiarity with C++ functions and control structures. You'll also need to do fundamental algorithmic error checking. These programs have to be tested on linprog before they are turned in.

Email your files `difference.cpp` and `evaluator.cpp` to `jayarama@cs.fsu.edu`

2 Program 1 - Special Difference

For this program, we define a new term called Special Difference. The Special Difference of a number is the difference between the number and the one you obtain by reversing its digits. For example, the Special Difference of 1234 is 3087 (| 1234 - 4321 |). In this program, you are required to find the sum of the special differences of a set of numbers. Make sure you conform to the following requirements.

1. Write a function called `reverse` that takes a number as a parameter and returns the reversed number. (20 points)
2. Write a function called difference that accepts a number as a parameter, calculates the special difference of the number and returns it. (15 points)
3. In the main function, accept a series of numbers from the user. Stop if the number entered is negative. Use the difference method to find the sum of the special differences and print it. (10 points)
4. Make sure you add comments to explain your logic. (5 points)

2.1 Sample Run

```
Enter the numbers:
1234
57
982
603
-2
The Sum is 4095
```

3 Program 2 - Taylor Series Expansion

The Taylor's Series expansion of e^x is defined as

$$\sum_{n=0}^{\text{inf}} \frac{x^n}{n!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} \cdots$$

Write a C++ program to evaluate e^x by using the Taylor's series expansion. Make sure you confirm to the following requirements.

1. Write a function called `factorial` that returns the factorial of a number. (15 points)
2. Write a function called `sum` that takes 'x' and a number 'n' as a parameter and calculates e^x accurate to the 'nth' term. (25 points)
3. You can use the `pow` function to calculate x^n.
4. In the main function, accept the values of 'x' and 'n' from the user and use the `sum` function to calculate the required value and print it. (5 points)
5. Make sure you add comments to explain your logic. (5 points)

3.1 Sample Run

```
Enter the value of x: 5
Enter the number of terms: 20

e^x is 148.413
```

4 Extra Credit Problem - Palindromic Primes

For this problem, you need to adapt your prime number program to check for palindromic primes. Palindromic primes are defined as prime numbers that are the same if read forwards or backwards. For example, 757 is a palindromic prime, but 59 is not. You need to print all the palindromic primes below a certain upper limit.

1. This program is worth 50 points, and is called `palPrimes.cpp`. Turn this in with the other programs.
2. Use the function you wrote for the previous program to reverse a number. (5 points)
3. Accept the upper limit from the user. If the limit is 0, 1 or negative, terminate the program(5 points)
4. Generate a prime number. Check if it is a palindrome. If it is, print it. If not, move on to the next number. (35 points)
5. Please include comments wherever appropriate. (5 points)

4.1 Sample Runs

```
Please enter the upper limit:  250
The palindrome primes are:
2
3
5
```

```
7
11
101
131
151
181
191
```

5 Generic Guidelines

1. Please make sure you're only using the concepts already discussed in class. That is, please try and restrict yourself to input/output statements, variables, selection statements and loops, and functions. Do not use arrays.
2. Each program is worth 50 points.
3. Please make sure that you're conforming to specifications (program name, print statements, expected inputs and outputs etc.).
4. Please make sure your code is readable.
5. Please make sure you've compiled and run your program before you turn it in. Compilation errors can be quite costly.