COP 3014: Fall 2021
Homework 4

Total Points: 100 (plus 50 point extra credit)
Due: Monday 10/18/2021 11:59:00 PM EDT

1 Objective

The objective for this assignment is to make sure

- You are comfortable with C++ functions (pre-defined and user-defined).
- You are familiar with C++ control structures (loops and branches) in any combination.
- You can recover from wrong user inputs.
- You are comfortable with console input and output involving numeric types.
- You can approach a complex problem, break it down into various parts, and put together a solution.

For this assignment, please make sure you conform to Output requirements. You should now be familiar enough with cout statements and output formatting that you should be able to EXACTLY match the sample output (other than certain exceptions you will be informed about beforehand).

You also need to match function and program names EXACTLY as specified. Not doing so will result in a loss of points.

This assignment requires you to submit multiple files on Canvas. Please do so in a SINGLE submission. Canvas allows you to turn in multiple flies in one submission. Once you have uploaded your first file, click "attach another file" to upload your second file, and so on.

Turn in your files `difference.cpp` and `evaluator.cpp` (and the extra credit program `palnums.cpp`, if you are attempting it) through Canvas.

2 Program 1 - Special Values

Patrick Star wasted a lot of time in Boating School instead of signing up for his Spanish Class. Unfortunately for him, Spanish 101 is now full, and the only other class that will suit his schedule is Advanced Math. Patrick is determined not to let this defeat him. He will make his way up the stairs of learning one way or another. However, it's been a while since Regular Math, and he is somewhat behind. Patrick is going to put his programming class (he took that last term) to good use and write programs to do his homework. For the rest of this homework, you are Patrick Star, trying to outwit the math teacher.

For this program, we define a new term called Special Value. The Special Value of a number is the difference between the one you obtain by reversing the digits of the given number and a randomly generated number *mod* the original number. For example, the Special Difference of 1234 could be 3934 (| 4321 - random % 1234 |). In this program, you are required to find the sum of the special values 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. (17 points)
2. Write a function called `value` that accepts a number as a parameter, calculates the special value of that number, and returns it. This function should call the `reverse` function. (15 points)
3. In the main function, first initialize your RNG by accepting a seed from the user (3 points)
4. Then, accept a series of numbers from the user. Stop if the number entered is 0. Use the `value` function to find the sum of the special values and print it. (10 points)
5. Make sure you add comments to explain your logic. (5 points)
6. You're only allowed the `iostream` and `cstdlib` libraries.
7. Seeding the RNG ensures 2 runs of the program on the same machine would produce the same output, so you may test your output by hand if required.
8. If you have questions about the RNG, please contact the instructional staff.

2.1 Sample Run

Please note that the final answer depends on the random number generated at each function call, and you might get a different answer.

```
Enter the seed: 36594
Enter the numbers:
1234
57
982
603
0
The Sum is 5183
```

3 Program 2 - Mclaurin Series Expansion

For the second homework, you (Patrick) are required to Mclaurin 's Series expansions for the trigonometric sine functions. The Mclaurin 's Series expansion of sin(x) is defined as

$$sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots$$

Write a C++ program to evaluate sin(x) by using the Mclaurin 's series expansion. For this program, you should not use the `sin` function in the cmath library. Doing so would result in a loss of most of the points for this program. You need to calculate the Mclaurin 's series expansion. Make sure you conform to the following requirements.

1. Write a function called `factorial` that returns the factorial of a number. (10 points)
2. Write a function called `sum` that takes 'x' and a number 'n' as a parameter and calculates sin(x) accurate to the 'nth' term. (25 points)
3. You can use the `pow` function to calculate x^i.

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. You may assume that ‘n’ will always be a positive number.

6. You may assume that ‘x’ will only ever be in the range (-3.14,3.14) which is the range for which the sine function is defined.

7. Print the margin of error, which is the difference between our function and cmath’s sine function for the same ‘x’. Make sure the answer is accurate to 6 digits after the decimal point. (5 points)

8. Make sure you add comments to explain your logic. (5 points)

9. You are only allowed to use the `cmath, iostream` and `iomanip` libraries.

3.1 Sample Run 1

```
Enter the value of x: -2.849
Enter the number of terms: 21
sin(-2.849000) = -0.288436
Margin of error: 0.000000
```

3.2 Sample Run 2

```
Enter the value of x: 3.001
Enter the number of terms: 6
sin(3.001000) = 0.140141
Margin of error: 0.000011
```

4 Extra Credit Problem - Palindromic Numbers

For this problem, you need to write a program to check for palindromic numbers. Palindromic numbers are defined as numbers that are the same if read forwards or backwards. For example, 757 is a palindromic number, but 259 is not. You need to print all the even palindromic numbers of a certain number of digits that are also multiples of a given number.

1. Use the function you wrote for the previous program to reverse a number. (5 points)

2. Write a function called `printPalin`. This function accepts 2 integer parameters - the number of digits and a factor, and returns nothing. The function loops through all numbers with the given number of digits and prints the palindromic numbers that are even and a multiple of the second parameter. (25 points).

3. In the main function, accept the number of digits from the user. If this number is smaller than 1 or greater than 9, print an error message. (5 points)

4. Accept a factor from the user. All numbers printed should be multiples of this number. If the given number if 1 or smaller, print an error message. (5 points).

5. If both inputs are acceptable, call the `printPalin` function. (5 points)

6. Please include comments wherever appropriate. (5 points)

4.1 Sample Run 1

```
Enter the number of digits: 7
Enter the factor: 10
The even 7 digit palindrome numbers that are multiples of 10 are:
```

4.2 Sample Run 2

```
Enter the number of digits: 5
Enter the factor: 32
The even 5 digit palindrome numbers that are multiples of 32 are:
21312
23232
25152
25952
27072
27872
29792
40704
42624
44544
46464
48384
61216
63136
63936
65056
65856
67776
69696
80608
82528
84448
86368
88288
```

4.3 Sample Run 3

```
Enter the number of digits: 11
The number is not in the expected range (1-9)
Enter the factor: -3
The number is not in the expected range (2 or higher).
```

5 Generic Guidelines

1. Include the header comment with your name and other information on the top of your files.
2. Please make sure you're only using the concepts already discussed in class. Please restrict yourself to input/output statements, variables and operators, selection statements, loops and functions. Using strings, arrays or anything more advanced will result in a loss of 20 points.
3. Each program is worth 50 points.

4. If we have listed a specification and allocated point for it, you will lose points if that particular item is missing from your code, even if it is trivial.

5. No global variables (variables outside of main())

6. No use of the `auto keyword` or any other C++ 11 or higher features.

7. Functions have to be declared above main and defined below main. Not doing so will result in a loss of 5 points per function.

8. You need to follow the other programming conventions established in the course.

9. **This is individual work. You may NOT collaborate with other students in the course, former students, hire tutors to "help", copy solutions off the internet, or use pay-for solution websites, including but not limited to Chegg, CourseHero, WiseAnt, Bartelby, tutor.com, assorted Social Media groups, whatever GroupMe students might have created, etc.) This includes posting the problem statements on these websites even if you do not use the answer obtained. Doing so is a violation of the Academic Honor Code. Violation of the Honor Code will result in a 0 grade on the program, a reduced letter grade in the course and potentially more serious consequences.**

10. All input and output must be done with streams, using the library `iostream`

11. You may only use the `iostream`, `iomanip`, `cmath` and `cstdlib` libraries (you do not need any others for these tasks)

12. Please make sure that you're conforming to specifications (program name, function names, parameters, return types, print statements, expected inputs and outputs etc.). Not doing so will result in a loss of 10 points.

13. NO C style printing is permitted. (Aka, don't use printf). Use cout if you need to print to the screen.

14. When you write source code, it should be readable and well-documented (comments).

15. Make sure you either develop with or test with Jetbrains CLion (to be sure it reports no compile errors or warnings) before you submit the program.

16. Testing your program thoroughly is a part of writing good code. We give you sample runs to make sure you match our output requirements and to get a general idea of how we would test your code. Matching your outputs for JUST the sample runs is not a guarantee of a 100. We have several extensive test cases.

17. Please make sure you've compiled and run your program before you turn it in. Compilation errors can be quite costly. We take 5 points off per compiler error for the first 9 errors. The 10th compiler error will result in a grade of 0.

18. Only a file turned in through Canvas counts as a submission. A file on you computer, even if it hasn't been edited after the deadline, does not count.

19. The student is responsible for making sure they have turned in the right file(s). We will not accept any excuses about inadvertently modifying or deleting files, or turning in the wrong files.

20. **Program submissions** should be done through the Canvas class page, under the assignments tab (if it's not there yet I'll create it soon.) Do not send program submissions through e-mail – e-mail attachments will not be accepted as valid submissions.

21. The ONLY files you will submit via Canvas are `difference.cpp`, `evaluator.cpp` and `palnums.cpp`

22. **General Advice** - always keep an untouched copy of your finished homework files in your email. These files will have a time-stamp which will show when they were last worked on and will serve as a backup in case you ever have legitimate problems with submitting files through Canvas. Do this for ALL programs.