COP 3363: Spring 2021- Homework 4

Total Points: 100 (and 50 points extra credit)

Due: Thursday, 03/11/2021, 11:59:00 PM

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 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). Not doing so will result in a loss of 5 points.

You also need to match function and program names EXACTLY as specified. Not doing so will result in a loss of 5 points.

Turn in your files FSUID_flipSum.cpp and FSUID_probability.cpp (and the extra credit program FSUID_palnums.cpp, if you are attempting it) using the submission script through linprog. Here, "FSUID" is your FSUID (Canvas login). For example, abc17z_flipSum.cpp.

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 Average Everyday 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 assignment, you are Patrick Star, trying to outwit the math teacher.

The numbers used for Patrick's exercise are all printed in the font shown below:



Printed in this form, numbers consisting of the digits 0, 1, 2, 5, 6, 8, and 9 can be "flipped". That is, they are intelligible the right side up, and upside down. For this assignment, we need to find the sums of a number and it's flipped counterpart. For example, the "flipped sum" of 1206 is 1206 + 9021 = 10227. Make sure you conform to the following requirements.

- 1. Call this program FSUID_flipSum.cpp
- 2. Write a function called flip that takes an integer as a parameter and returns the flipped number. (20 points)
- 3. Write a function called value that accepts a number as a parameter, calculates the flipped sum of that number, and returns it. This function should call the flip function. (10 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 flipped sum of each of the numbers as they are entered, print them. Also, calculate the average flipped sum and print it. (15 points)
- 5. Make sure you add comments to explain your logic. (5 points)
- 6. You are only allowed the iostream library for this problem.
- 7. Please note that inputs will not have leading 0's, but the flipped numbers might. For example, 1200, when flipped is 0021, which is the number 21.

2.1 Sample Run 1

Enter the numbers (0 to stop): 128 The flipped sum of 128 is 949 5069 The flipped sum of 5069 is 11974 9125 The flipped sum of 9125 is 14341 200 The flipped sum of 200 is 202 0 The average flipped sum is 6866

2.2 Sample Run 2

```
Enter the numbers (0 to stop):
0
No valid data entered
```

3 Program 2 - Poisson Mean through Series expansion

Mr. Krbas, now aware of Patrick's increasing reputation as a mathematician (that is willing to work for his offered salary), has hired him to calculate if business at the Krusty Krab is at the optimum level. This can be modeled on a Poisson distribution. You (Patrick) needs to calculate the cumulative Distribution function of the Poisson distribution using the formula:

$$F(x,\lambda) = e^{-\lambda} \sum_{k=0}^{x} \frac{\lambda^k}{k!} = e^{-\lambda} \left(\frac{\lambda^0}{0!} + \frac{\lambda}{1!} + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!} + \dots + \frac{\lambda^x}{x!} \right)$$

Write a C++ program to evaluate the cumulative distribution function of the Poisson Distribution. For this program, you may assume the value of e is 2.71828. Make sure you conform to the following requirements.

- 1. Call this program FSUID_probability.cpp
- 2. Write a function called factorial that returns the factorial of a number. (10 points)
- 3. Write a function called sum that takes 'x' and ' λ ' as parameters and calculates F(x, λ) (25 points)
- 4. Oversimplified definition: 'x' can be considered to represent the number of customers in the restaurant at a time, and ' λ is analogous to the expected number of customers. The CDF then can represent the probability having less than the expected number of customers at a time.
- 5. You may use the pow function to raise a number to a power.
- In the main function, accept various values of 'x' and 'λ' from the user and use the sum function to calculate the required CDF probability and print it. Stop if the user enters a negative value for x. (8 points)
- 7. If the CDF for a set of inputs was less than 0.5, then the restaurant is not at optimum (either due to too few or too many customers). If is is so, print it. (2 points).
- 8. You may assume that 'x' will always be an integer.
- 9. You may assume that λ will always be a positive number.
- 10. You may use the iostream, iomanip and cmath libraries.
- 11. Make sure you add comments to explain your logic. (5 points)

3.1 Sample Run

Enter the expected number of customers: 25 Enter the actual number of customers : 10 Probability of less than 10 customers was 0.0006 Under optimum performance Enter the actual number of customers : 35 Probability of less than 35 customers was 0.9776 Enter the actual number of customers : 19 Probability of less than 19 customers was 0.1336 Under optimum performance Enter the actual number of customers : 27 Probability of less than 27 customers was 0.7002 Enter the actual number of customers : -5

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. Call this program FSUID_palnums.cpp
- 2. Modify the function you wrote for the previous program to reverse a number. (5 points)

- 3. 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).
- 4. 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)
- 5. 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).
- 6. If both inputs are acceptable, call the printPalin function. (5 points)
- 7. Please include comments wherever appropriate. (5 points)
- 8. You may only use the iostream library.

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	e ni	umber	of	digits:	5						
Enter	the	e fa	actor	: 3	2							
The e	ven	5 d	ligit	pa	lindrome	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: -3The number is not in the expected range (2 or higher).

5 Submission

You will be submitting the file directly through the programming server. Please follow these instructions:

- 1. Write and test your program on the server. Please name your file properly. The "FSUID" in the example below should be replaced with your FSUID (Canvas login).
- 2. You need to submit the 3 files separately. In the next 2 commands, replace "filename" with either "flipSum.cpp", "probability.cpp" or "palnums.cpp".
- 3. When you're ready to submit, first change the file permissions to give us read permissions: chmod 744 FSUID_filename
- Run the submission script: ~jayarama/homework/submitHW4 FSUID_filename

Please note the following:

- 1. THE FILE HAS TO BE NAMED CORRECTLY. You run the risk of overwriting other files
- 2. WE ONLY NEED YOUR CPP FILE. Please don't submit the compiled executable.
- 3. IF YOU GET THE "File Submitted" MESSAGE, YOU HAVE SUCCESSFULLY SUBMITTED THE FILE.
- 4. IF YOU MAKE MULTIPLE SUBMISSIONS, THE OLDER SUBMISSIONS WILL BE DELETED. The script will only keep the latest version of your file, and that is the submission we'll grade.

6 General Requirements

- 1. Please name your file according to the given naming convention, including case. Incorrect file names will result in a loss of 3 points.
- 2. Include the header comment with your name and other information on the top of your files.
- 3. Please make sure you're only using the concepts already discussed in class. Please restrict yourself to variables, operators, selection statements, loops and functions. Using strings, arrays, C++ 11 and up features or anything more advanced will result in a loss of 20 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. All input and output must be done with streams, using the library iostream
- 7. You may only use the libraries indicated in each problem (you do not need any others for these tasks). Use of other libraries will result in a loss of 10 points per library.
- 8. Each program is worth 50 points.

- 9. Functions have to be declared above main and defined after main. Not doign so will result in a loss of 10 points per program.
- 10. NO C style printing is permitted. (Aka, don't use printf). Use cout if you need to print to the screen.
- 11. When you write source code, it should be readable and well-documented (comments).
- 12. Make sure you either develop with or test with g++ on linprog (to be sure it reports no compile errors or warnings) before you submit the program.
- 13. 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.
- 14. 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.
- 15. Only a file turned in through the submission script counts as a submission. A file on your computer or your linprog account, even if it hasn't been edited after the deadline, does not count.
- 16. 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.
- 17. **Program submissions** should be done through the submission script mentioned above. Do not send program submissions through e-mail e-mail attachments will not be accepted as valid submissions.
- 18. The ONLY files you will submit are FSUID_flipSum.cpp, FSUID_probability.cpp and FSUID_palnums.cpp where "FSUID" has been replaced with your FSUID (Canvas login).
- 19. 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 the submission script. Do this for ALL programs.

Spongebob Squarepants and related works are the property of Nickelodeon and are used here as examples for educational purposes only. This is covered under Fair Use.