CPS 210 Lab 6

PROBLEMS

For 1 through 7, use ***both*** a for loop and a while loop. For the rest of the questions, choose whether to use a for or while loop.

1. Print 50 to 10.

2. Print 0 to 100.

   The output is kind of hard to read because it's so long. Even if you printed one per line or all on one line.

   Try to print 0 to 100 with 10 numbers per line. **Hint:** try using modulus %

3. Print the alphabet 'A' to 'Z'.

   - **Hint:** You can use **char** instead of int in your initialization statement.

4. Print the alphabet backwards 'z' to 'a'.

5. Print the sum of the even, positive integers less than 50.

6. Count the numbers divisible by 2 **or** 7 between 20 to 300 inclusive.

   - **Remember:** OR in Java is ||, AND is &&

7. Count the number of odd numbers between 15 and 75 inclusive.

8. Write a program that displays the following table. Kilograms should start from 1 to 199 and be odd.

   - **Note:** 1 kilogram is 2.2 pounds):

Kilograms	Pounds
1	2.2
3	6.6
...	
197	433.4
199	437.8

9. Write a program to compute the sum of digits of any length integer. Use Scanner to obtain the integer from the user.

10. Remember our factorial question from lab 2?

Factorial represented in mathematics by the symbol **!** is the product of 1 to n. For example:

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Use a loop to make a program compute n! = 1*2*3*…n. Use Scanner to obtain n from a user.

11. One way pi can be approximated is by the following summation:
Pi = 4∗(1−1/3+1/5−1/7+1/9−1/11+...)

    Write a program to approximate pi using the first 20 terms of the summation above.

    Test your program with 20 terms, 200 terms, 2000 terms, 20000 terms.

    Pi= 3.14159265359

    Notice that the more terms you sum the more accurate the value estimates pi. This is a tricky question. Think about how you can go about going back and forth to adding and subtraction every loop.

12. The following question is an example of when using a while loop is convenient since we don't know how many times this loop will run.

Sections 5.2–5.7

***5.1** (*Count positive and negative numbers and compute the average of numbers*) Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input **0**. Display the average as a floating-point number. Here is a sample run:

```
Enter an integer, the input ends if it is 0: 1 2 -1 3 0 ↵ Enter
The number of positives is 3
The number of negatives is 1
The total is 5.0
The average is 1.25
```

```
Enter an integer, the input ends if it is 0: 0 ↵ Enter
No numbers are entered except 0
```