CS 271
Lab Assignment # 3

Submit **lab3.c, lab3functions.c,lab3functions.h, and makefile** on Canvas Lab 3 Assignment.

Read Chapter 5.

Pay particular attention to the features of C that are different from Java:
Function prototypes
Argument coercion
Random number generation

Warning: This lab assignment requires Linux. If you have Linux installed on your own computer and you have the GNU gcc compiler, you may try using your own computer but you should still test your files on the computer science machines (SH 118 or 118B) before submitting.

1. Make a new directory for this lab. Place all of the files you create for this lab assignment in that directory.

2. Create a header file named lab3functions.h. In this file, place the following lines:

```
#ifndef LAB3FUNCTIONS
#define LAB3FUNCTIONS
        here is where your function prototypes go
#endif
```

3. Create a source file named lab3functions.c In this file, place the implementations of the following three functions:

   - A function named `perfect` that will accept an integer parameter. The function should return 1 if the parameter is a perfect number. Otherwise, it should return 0.

     An integer is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3.

   - A function named `prime` that will accept an integer parameter. The function should return 1 if the number is prime. Otherwise, it should return 0. (See the pseudocode below)

     function prime (int n)
       loop from x = 2 to x = sqrt(n)
         if n is divisible by x, return 0
       end loop
       return 1
     end function

- A function named revDigits that accepts an integer parameter. The function should return the number with its digits reversed.

  Example: revDigits(7631) should return 1367

  Example: revDigits(1840) should return 481

  Example: revDigits(-945) should return -549

▸▸ Remember to put `#include "lab3functions.h"` at the top of the file.

4. Create a test program named lab3.c. In this file, place the main function as follows:

```
// header comments
#include "lab3functions.h"
#include <stdio.h>
int main (void) {
   // use the perfect function to print the positive, perfect numbers
   // less than 1000

   // use the prime function to print the first 20 positive, prime numbers
   // note:  1 is not a prime

   // use the revDigits function to print the reverse of several (at least 3)
   // numbers with various numbers of digits
}
```

▸▸ Any .c file must have header comments and proper indentation.

▸▸ Remember to put `#include "lab3functions.h"` at the top of the file.

5. Compile lab3functions.c to make an object file. Debug the program and repeat until it compiles successfully. (At this point, we just want to know that there are no syntax errors.)

```
gcc -c lab3functions.c
```

6. Compile lab3.c to make an object file. Debug the program and repeat until it compiles successfully. (At this point, we just want to know that there are no syntax errors.)

```
gcc -c lab3.c
```

7. Create a makefile. There is a separate instruction sheet for this. The "all" target dependency should be named **lab3** (all lowercase).

8. "make" lab3. If the makefile works correctly, run the executable.

Submit 4 files:

1) makefile
2) lab3functions.h
3) lab3functions.c
4) lab3.c