CSE 165 - LAB 02

Points: 100

Write a separate .cpp file for each of the following tasks.

1. If-else statements (10 points)

Write a program that reads in an integer and outputs "POSITIVE" if the integer is positive and "NEGATIVE" if it is negative, and "ODD" if the integer is odd and "EVEN" if the integer is even.

2. While loops (10 points)

Modify the program just you wrote for the previous exercise so that it keeps reading integers and classifying them as ODD or EVEN and POSITIVE or NEGATIVE until the user inputs 0. Then the program ends. No classification should be produced for the 0.

3. Finding Prime Numbers (20 points)

Write a program that reads in an integer N and prints out all the prime numbers strictly less than N. These should be printed one per line.

4. Formatted Output (20 points)

Write a program that keeps reading in positive integers and for each integer outputs the corresponding value in hexadecimal and binary format. Stop when -1 is read. No output should be generated for -1.

5. Pointer Arithmetic (20 points)

Write a program where a user enters two integers. Create a pointer to each of the numbers. Multiply, add, subtract and divide the numbers by only using their pointers (remember to dereference them) and output the result to the console.

6. Bit Manipulation (20 points)

Write a program that reads in a decimal number "n". Convert the decimal number into binary format and print out the number in binary format. Then, write the following functions to get, set and clear bit at the position "index" and display the corresponding output given below.

```
/*
 Retrieve a bit from a number "n" in binary format at position "index"
 Input: number n, position index where 0 being the right most(least significant) bit
 Output: bit at position "index"
 Example: Input: n=10 (in binary “1010”), index=0, output: 0
*/
int getBit(int n, int index)

/*
 Set a bit at position "index". This will set a bit to 1 if its 0 or else it will
 remain unchanged
 Input: number "n", position "index" where 0 being the right most(least significant) bit
 Output: Output the binary number after a bit is set at position "index"
 Example: Input: n=10 (in binary “1010”), index=0, output: 11 (in binary “1011”)
*/
int setBit(int n, int index)
```

```
/*
 Clear a bit at position "index". This will set a bit to 0 if it is 1 or else it
 will remain unchanged.
 Input: number "n", position "index" where 0 being the right most(least significant) bit
 Output: Output the binary number after a bit is cleared at position "index"
 Example: Input: n=10 (in binary “1010”), index=2, output: 10 (in binary “1010”)
*/
int clearBit(int n, int index)
```

Instruction for submission:

1. Zip all your .cpp files together
2. Submit the zip folder to catcourses