CSE-165 Lab 02

Points: 100

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

Zip all your .cpp files together and submit the zip file to CatCourses.

---

1. **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.

---

2. **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 a negative number is read. No output should be generated for the negative number.

---

3. **Pointer Math (20 points)**

Write a program that takes two integers as input. Create a pointer to each of the numbers. Add, subtract, multiply, and divide the numbers together using the pointers (remember to dereference them) and output the result to the console. Note: the result of the division should be a float.

CSE-165 Lab 02

Points: 100

4. **Bit Manipulation (40 points)**

a) Write a program that reads in a decimal number "n" and a position "index". Convert the decimal number into binary format and print out the number in binary format.

b) Next, implement the following functions to get, set, and clear the bit at the position "index" of the number.

```
/*
 Retrieve a bit from a number "n" in binary format at position "index"
 Input: number n, position index with 0 being the right most (least
significant) bit
 Output: bit at position "index"
 Example: Input: n=1010, index=1, output=1
*/
int getBit(int n, int index)
```

```
/*
 Set a bit at position "index"
 Input: number n, position index with 0 being the right most (least
significant) bit
 Output: the binary number after the bit is set at position "index"
 Example: Input: n=1010, index=0, output=1011
*/
int setBit(int n, int index)
```

```
/*
 Clear a bit at position "index"
 Input: number n, position index with 0 being the right most (least
significant) bit
 Output: the binary number after the bit is cleared at position "index"
 Example: Input: n=1010, index=1, output=1000
*/
int clearBit(int n, int index)
```

c) Finally, using your functions, print the following three things: 1) the original bit at position "index" of "n"; 2) the number "n" with the bit at "index" set; 3) the number "n" with the bit at "index" cleared. Essentially, use the two inputs to the program to call your three functions and print their returns.

Note: To display a decimal number in 8-bit binary format, you can use `std::bitset<8>(n)`.

For example: `std::bitset<8>(4)` will output 00000100.

Example Inputs: 10 and 2

Output: Binary representation of 10 is 1010

Get bit at index 2: 0

Binary number after setting bit at index 2: 1110

Binary number after clearing bit at index 2: 1010

Example Inputs: 547 and 8

Output: Binary representation of 547 is 1000100011

Get bit at index 8: 0

Binary number after setting bit at index 8: 1100100011

Binary number after clearing bit at index 8: 1000100011

Example Inputs: 1023 and 5

Output: Binary representation of 1023 is 1111111111

Get bit at index 5: 1

Binary number after setting bit at index 5: 1111111111

Binary number after clearing bit at index 5: 1111011111