

Assignment 3

DUE Thursday, October 6, 2016 by 11:59.59.99PM (before MIDNIGHT)

Submit your file to your Canvas “Homework ASSIGNMENT 3” assignment in the Homework ASSIGNMENTS group.

Preliminary Information:

You will be creating and submitting one file for this Assignment: A3.c.

Assignment 3 Program Description:

There are several parts to this assignment. You will need to create a menu, at least one function, and a `main()` or calling function, that is interactive with the user. Read carefully to be sure everything happens in the correct order and that all of your options yield the expected/required output.

REMEMBER: Your Main Menu should look **exactly the same** as the samples below, including the order of the items on the menu, and the interactive input shown should result in **exactly the same output**, including **display formatting**, as the samples below. Be careful to note that all input will be in **base10**.

It may be easier to understand the functionality of the program by carefully examining the Sample Output, as it is a bit challenging to describe.

The Problem: Create an interactive program that performs 6 different bit-based operations, based on user input of one or two base10 integers, and outputs the results in both binary and base10 notation or representation.

The six (6) operations are:

1. **Left Bit Shift:** Input will be both a base10 integer and “number of places to shift” value.
2. **Right Bit Shift:** Input will be both a base10 integer and “number of places to shift” value.
3. **Binary AND:** Input will be two base10 integers.
4. **Binary OR:** Input will be two base10 integers.
5. **Binary Negation or “One’s Complement” (~):** Input will be a base10 integer.
6. **Binary XOR:** Input will be two base10 integers. **[If you are not sure about Exclusive OR, look it up.]**

Output for each operation will be of the form:

```
Entered Integer: [binary display] (base10 [base10 value])
Output          : [binary display] (base10 [base10 value])
```

BINARY DISPLAY must be accomplished by calling a function `showbits` (or **very** similar name) that takes an `int` as an argument and prints out the entire `int` in binary notation, i.e., “127” is the argument and the display is 01111111. **NOTE: Your output should always be 32 places, as in the sample, regardless of input, for full credit. Similarly, well-formatted output, as in the sample output, is required for full credit. You are also expected to use function prototypes wherever they would “apply.”**

NOTE: Any action that makes the left-most or most significant bit a “1” should be displayed in base10 as two’s complement. This should happen “by default.” You should note this in the sample output and your version of this program should do the same.

MAIN MENU must allow the user to select any of the 6 operations repeatedly until a 7th option is selected that exits the program. Make sure proper user prompts and instructions are displayed so anyone could use your version of this program without prior instruction.

SAMPLE OUTPUT BEGINS ON NEXT PAGE.

Sample Output for all 6 operations and "Quit":

[jwhitmer@silo.cs.indiana.edu] bittest

Main Menu

1. Perform Left Bit Shift Operation (one value)
 2. Perform Right Bit Shift Operation (one value)
 3. Perform Binary AND Operation (two values)
 4. Perform Binary OR Operation (two values)
 5. Perform Binary One's Complement (~) Operation (one value)
 6. Perform Binary XOR Operation (two values)
 7. Quit
- 1

Enter a base 10 Integer: 1234

Enter number of places to shift bit: 8

Entered Integer: 00000000000000000000000010011010010 (base 10 1234)
Left Shifted : 00000000000000001001101001000000000 (base 10 315904)

Main Menu

1. Perform Left Bit Shift Operation (one value)
 2. Perform Right Bit Shift Operation (one value)
 3. Perform Binary AND Operation (two values)
 4. Perform Binary OR Operation (two values)
 5. Perform Binary One's Complement (~) Operation (one value)
 6. Perform Binary XOR Operation (two values)
 7. Quit
- 2

Enter a base 10 Integer: 1234

Enter number of places to shift bit: 8

Entered Integer: 00000000000000000000000010011010010 (base 10 1234)
Right Shifted : 00000000000000000000000000000000100 (base 10 4)

Main Menu

1. Perform Left Bit Shift Operation (one value)
 2. Perform Right Bit Shift Operation (one value)
 3. Perform Binary AND Operation (two values)
 4. Perform Binary OR Operation (two values)
 5. Perform Binary One's Complement (~) Operation (one value)
 6. Perform Binary XOR Operation (two values)
 7. Quit
- 3

Enter two base 10 Integers, separated by a space: 1234 5678

Entered Integer 1: 00000000000000000000000010011010010 (base 10 1234)
Entered Integer 2: 0000000000000000000000001011000101110 (base 10 5678)
AND'ed : 00000000000000000000000010000000010 (base 10 1026)

- 4

[illegible]

- 5

```
Entered Integer: 00000000000000000000010011010010 (base 10 1234)
~'ed           : 111111111111111111111101100101101 (base 10 -1235)
```

- 6

A290/A590
System Programming with C and Unix
Assignment 3

Enter two base 10 Integers, separated by a space: 1234 5678

```
Entered Integer 1: 00000000000000000000000010011010010 (base 10 1234)
Entered Integer 2: 0000000000000000000000001011000101110 (base 10 5678)
XOR'ed           : 0000000000000000000000001001011111100 (base 10 4860)
```

Main Menu

```
1. Perform Left Bit Shift Operation (one value)
2. Perform Right Bit Shift Operation (one value)
3. Perform Binary AND Operation (two values)
4. Perform Binary OR Operation (two values)
5. Perform Binary One's Complement (~) Operation (one value)
6. Perform Binary XOR Operation (two values)
7. Quit
7
[jwhitmer@silo.cs.indiana.edu]
```

Assignment 3 General Program Requirements:

Write your `main()` or “calling function” so the Main Menu functions as displayed/indicated by the sample output and each of the 6 operations accepts the indicated input and produces the indicated output. **MAKE SURE** your display of the binary representation is controlled by calling a function specifically to create that binary display.

Assignment 3 **must** include the following features:

1. Proper Comment Block at the head of the file.
2. Proper and appropriate “in-line” or “in the code” comments.
3. Input and output as indicated in each operation.
4. Properly written `main()` or “calling function” to test all operations **AND** generate binary display output using called function “**showbits**” or equivalent.
5. Proper use of function prototypes.
6. Your program compiles for testing.

Again, **BE SURE** you thoroughly comment your actual code and include the expected Heading Comment Block. These comments and comment block will represent 20% of your score for this Assignment.

Scoring:

Proper Comments and Heading Comment Block: **20 points**

File compiles: **20 points** and `main()` is written to properly test all operations via Main Menu using called binary print function, i.e. **showbits**. [**showbits** is worth 10 of these 20 points]

File compiles and accepts proper input and generates proper output: **60 points distributed** [10 points each] across 6 functions as indicated above.

Handing in your Assignment.

As with all other submitted programs/assignments, you will have to use some form of a “Secure FTP” program. It will usually take two steps to submit to Canvas: 1: Move the file from silo to the computer you are using, 2. Upload the file from the computer you are using to Canvas with the usual method. If you have questions about this, ask them ASAP.