

## **Assignment 3**

### **CISC 220, Fall 2018**

#### **Administrative Notes:**

All of the administrative notes from previous assignments apply to this assignment as well. I will just summarize the main points here and you can refer back to the previous assignments instructions for more details:

- No group work allowed.
- Hand in your file to OnQ in the “Assignment 3” Dropbox area.
- Follow instructions carefully or you may lose points.
- There is a 24-hour “grace period” after the official deadline for each assignment; submissions made within this period lose 20% but are still accepted.

#### **Deadline:**

The deadline for this assignment is **Tuesday, October 23 at 8 a.m.** The grace period lasts until Wednesday, October 24 at 8 a.m.

#### **Required Program:**

This is your first C assignment and your task is to write a fairly simple C program. It can be done with the features of C we've discussed in the “Introduction to C” topic. The program requires interacting with users to collect certain information. Then produce an output based in the given input. You may use what we have covered and explained in the class so far. So you don't have to use anything from the next topic (“Pointers, Arrays & Strings”).

The object of your program is to demonstrate the skills you have learned in C programming. Your program basically will use only 3 Bitwise operators.

Here is what your program must do:

1. Prompt the user to enter a number for an “A” variable between 5 and 50.
2. Prompt the user to enter a number for a “B” variable between 5 and 50.
3. Prompt the user to choose which Bitwise operator to use for the given “A” and “B” value. e.g:  
Enter 1 for AND  
Enter 2 for OR  
Enter 3 for XOR
4. Output the result. e.g. 12 OR 25 = 2
5. Prompt for “A” and “B” again .... etc. (repeat 1 to 4 above)
6. Stop prompting when the user enters “-1” to signal the end.

## Error Checking:

The program should detect the following errors:

1. If the user enters an alphabet character or any other special characters, your program must write an error message “You enter an alphabet character(s), enter only a digit character(s)”
2. If the user enters a digit character(s) that is outside the give range (5 – 50), your program must write an error message “Your number must be between 5 and 50.”
3. If the user enters “A” and “B” correctly, and proceed to choosing one of three Bitwise operators. But enters a number that is not 1, 2, or 3, the program should produce an error message indicating that only “1, 2, or 3” are allowed.
4. If your program generates errors as 1, 2 above, it should continue and prompt the user again for “A” value, then “B” ... etc. If the program run into error 3 above, it should continue and prompt the user again to enter only “1, 2, or 3.”

Also, your error messages must be clear and say what's wrong (i.e. not just “error”) but they don't have to be word-for-word the same error messages as shown in the sample runs included at the end of this document. Use a regular `printf` for your error messages.

## Use of Libraries:

Mainly, you should include at least `<stdio.h>` in your program.

## More Requirements:

These additional requirements are designed to make sure you're practicing several features of C:

- Your program must include at least two `#defines` and use the symbols in a non-trivial way in the program. For example, you could use `#defines` for the minimum and maximum allowed values, and the special sentinel value -1 for the end of the user's input. The advantage of this is that if you later change your mind about one of these values you only have to change one line of the program instead of hunting through the whole thing to find places where you used the value. You can use some or all of these ideas and/or make up your own.
- Your program must be in a single file called `assn3.c`.

## Marking Scheme:

- Reading correctly “A” and “B” values and produce correct evaluation of AND operator: 2
- Reading correctly “A” and “B” values and produce correct evaluation of OR operator: 2
- Reading correctly “A” and “B” values and produce correct evaluation of XOR operator: 2
- Reading the sentinel value “-1” and exit correctly at the “A” and “B” assignment stage: 1
- Reading the sentinel value “-1” and exit correctly at the “1, 2, and 3” stage: 1
- Displaying Error 1 for the value of “A”: 2
- Displaying Error 2 for the value of “A”: 2
- Displaying Error 1 for the value of “B”: 2
- Displaying Error 2 for the value of “B”: 2
- Displaying Error 3: 2
- Continue to prompt user for input until user enters “-1” : 1

- Using at least two #define's: 1

Total: 20

Note that the sentinel value can be effective at any possible point, i.e. at “A”, “B”, or “1, 2, or” stages. 标记值

If you do not follow the assignment requirements -- especially if you deviate from them in a way that costs us time while marking -- you will lose administrative points even if your program works correctly. So please read the directions carefully.

### **Possible Penalties:**

20% if you hand in during the late period (24 hours after the deadline).

10-20% if there are administrative issues that cause the graders extra time.

### **Style:**

Use tabs and proper indentation to organize your program. Use common naming conventions we have been using in our examples at classes.

### **Sample Run #1: No errors**

```
gcc -o assn3 assn3.c
enter a value of "A" : 12
enter a value of "B" : 25

enter 1 for A & B
enter 2 for A | B
enter 3 for A ^ B
enter your choice: 3

12 ^ 25 = 29
```

### **Sample Run #2: No errors**

```
gcc -o assn3 assn3.c
enter a value of "A" : -1
```

The program has been terminated.

### **Sample Run #3: error #1**

```
gcc -o assn3 assn3.c
enter a value of "A" : ABC
```

you have entered an alphabetic characters,  
enter a number between 5 and 50

### **Sample Run #4: error #1**

```
gcc -o assn3 assn3.c
enter a value of "A" : 12
enter a value of "B" : ABC
```

you have entered an alphabetic characters,  
enter a number between 5 and 50

**Sample Run #5: error #2**

```
gcc -o assn3 assn3.c
enter a value of "A" : 75
```

you have entered an a value out of accepted range,  
enter a number between 5 and 50

**Sample Run #6: error #2**

```
gcc -o assn3 assn3.c
enter a value of "A" : 12
enter a value of "B" : 75
```

you have entered an a value out of accepted range,  
enter a number between 5 and 50

**Sample Run #7: error #3**

```
gcc -o assn3 assn3.c

enter a value of "A" : 12
enter a value of "B" : 25
```

```
enter 1 for A & B
enter 2 for A | B
enter 3 for A ^ B
enter 1, 2, or 3: 5
```

Your entry is not accepted; it must be 1-3.

```
enter 1, 2, or 3:
```