This lab is an exercise in experimenting with *Bitwise Operations*, and writing code to manipulate the bits of an unsigned integer. **Do not be discouraged** if you need a significant amount of "trial-and-error" to get comfortable with bitwise operations. This is normal.

Table of Contents

Due Date

You must *submit* the source code for the solution to this lab exercise to *Moodle* by

Thursday, June 26, 2025

in order to receive full credit for this work. You must also *demonstrate* the working program to the instructor during class, within two weeks of submitting the source code. (In other words, **submit** the source code by the due date, and do not let yourself get too far behind regarding the demonstration of your working solution during class. This is the same policy that is described in the course *Syllabus*.)

Starter Code

For some **Lab Exercises**, the nature of the program required would be **too much to ask** students to write the program *entirely* on their own. In these situations, the assignment actually gives the student a program that ALREADY WORKS, but has one or more features that are only PARTIALLY complete. This Lab Exercise is an example of an assignment with STARTER CODE: Most of the actual code in the program **already works**. In fact, we encourage the student to study and understand the STARTER CODE, because it will help them understand the changes that they **do** need to make for the assignment.

Supported Commands

The “starter code” is a program that supports the following commands, except that the code for **two** of the commands is incomplete:

a	bitwise AND
c	CLEAR a particular hex digit (**to be completed by the student**).
d	DEFINE new contents for specific Hex digit. (**to be completed by the student**).
h	print help text.
l	bitwise LEFT SHIFT
n	NEGATE (invert) all bits.
o	bitwise OR
r	bitwise RIGHT SHIFT
s	SET data value
q	QUIT (end the program).
v	SET or CLEAR the “verbose mode” feature.
x	bitwise XOR
demo	demonstrate the use of “Verbose Mode” code (debug aids).

The code for each command calls a function that performs the task of that particular command.

Two-Part Lab Exercise

This Lab Exercise contains two distinct parts, that you should complete ***in order***:

1. **Manually Experiment** with the existing “Starter Code”, in order to get familiar with how *Bitwise Operations* work, and how to write code that uses *Bitwise Operations*.
2. **Programming Exercise:** complete the code to implement the ‘**c**’ and ‘**d**’ commands

Part 1: Manual Experimentation with Interactive Commands

Experiment with the commands to become familiar with how the program works.

The **Part 1: Sample Interactive Session** section (beginning on the next page) is a helpful guide to get you started as you experiment with the Starter Code.

The following table contains text that represents a possible execution session for the **STARTER** program. (In this example, what the user types is shown in **bold**. The purpose of this is to assist the reader of this document to distinguish text that the user entered from text that the program printed. When the program *actually* runs, text that the user types will appear in the same text format as the output.)

Sample Interactive Session (starter code)

Part 1: Sample Interactive Session (starter code)

```
Decimal:          0 Octal:           0 Hex:        0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): h
Supported commands:
     a    bitwise AND.
     c    Clear specific Hex digit. (Needs STUDENT modifications)
     d    Define new contents for specific Hex digit. (Needs STUDENT
modifications)
     h    Print this help text.
     l    (lower-case L) bitwise LEFT SHIFT
     n    Negate (invert) all bits.
     o    bitwise OR.
     r    bitwise RIGHT SHIFT.
     s    SET data value (hex).
     q    quit (end the program).
     v    SET or CLEAR verboseMode (debug aids).
     x    bitwise XOR.
   demo   Demonstrate how to use verboseMode.

 Decimal:          0 Octal:           0 Hex:        0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): s
Original data value (hex) = 0
Enter argument for new data value (hex): 5
Modified data value (hex) = 5
 Decimal:          5 Octal:           5 Hex:        5 Binary:
0000_0000_0000_0000_0000_0000_0000_0101
Enter command (or 'h' for help): l
Original data value (hex) = 5
Enter decimal argument for Bitwise-LeftShift: 2
Modified data value (hex) = 14
 Decimal:         20 Octal:          24 Hex:       14 Binary:
0000_0000_0000_0000_0000_0000_0001_0100
Enter command (or 'h' for help): o
Original data value (hex) = 14
Enter hex argument for Bitwise-OR: F00
Modified data value (hex) = F14
 Decimal:       3860 Octal:        7424 Hex:      F14 Binary:
0000_0000_0000_0000_0000_1111_0001_0100
Enter command (or 'h' for help): s
Original data value (hex) = F14
Enter argument for new data value (hex): ff0
Modified data value (hex) = FF0
 Decimal:       4080 Octal:        7760 Hex:      FF0 Binary:
0000_0000_0000_0000_0000_1111_1111_0000
Enter command (or 'h' for help): r
```

Part 1: Sample Interactive Session (starter code)

```
Original data value (hex) = FF0
Enter decimal argument for Bitwise-RightShift: 1
Modified data value (hex) = 7F8
 Decimal:       2040 Octal:        3770 Hex:      7F8 Binary:
0000_0000_0000_0000_0000_0111_1111_1000
Enter command (or 'h' for help): l
Original data value (hex) = 7F8
Enter decimal argument for Bitwise-LeftShift: 3
Modified data value (hex) = 3FC0
 Decimal:      16320 Octal:       37700 Hex:     3FC0 Binary:
0000_0000_0000_0000_0011_1111_1100_0000
Enter command (or 'h' for help): n
Original data value (hex) = 3FC0
Modified data value (hex) = FFFFC03F
 Decimal: 4294950975 Octal: 37777740077 Hex: FFFFC03F Binary:
1111_1111_1111_1111_1100_0000_0011_1111
Enter command (or 'h' for help): o
Original data value (hex) = FFFFC03F
Enter hex argument for Bitwise-OR: 300
Modified data value (hex) = FFFFC33F
 Decimal: 4294951743 Octal: 37777741477 Hex: FFFFC33F Binary:
1111_1111_1111_1111_1100_0011_0011_1111
Enter command (or 'h' for help): r
Original data value (hex) = FFFFC33F
Enter decimal argument for Bitwise-RightShift: 2
Modified data value (hex) = 3FFFF0CF
 Decimal: 1073737935 Octal:  7777770317 Hex: 3FFFF0CF Binary:
0011_1111_1111_1111_1111_0000_1100_1111
Enter command (or 'h' for help): x
Original data value (hex) = 3FFFF0CF
Enter hex argument for Bitwise-XOR: F0
Modified data value (hex) = 3FFFF03F
 Decimal: 1073737791 Octal:  7777770077 Hex: 3FFFF03F Binary:
0011_1111_1111_1111_1111_0000_0011_1111
Enter command (or 'h' for help): s
Original data value (hex) = 3FFFF03F
Enter argument for new data value (hex): 12345678
Modified data value (hex) = 12345678
 Decimal:  305419896 Octal:  2215053170 Hex: 12345678 Binary:
0001_0010_0011_0100_0101_0110_0111_1000
Enter command (or 'h' for help): a
Original data value (hex) = 12345678
Enter hex argument for Bitwise-AND: F0
Modified data value (hex) = 70
 Decimal:        112 Octal:         160 Hex:       70 Binary:
0000_0000_0000_0000_0000_0000_0111_0000
```

Part 1: Sample Interactive Session (starter code)

```
Enter command (or 'h' for help): l
Original data value (hex) = 70
Enter decimal argument for Bitwise-LeftShift: 2
Modified data value (hex) = 1C0
 Decimal:        448 Octal:         700 Hex:      1C0 Binary:
0000_0000_0000_0000_0000_0001_1100_0000
Enter command (or 'h' for help): x
Original data value (hex) = 1C0
Enter hex argument for Bitwise-XOR: FFFFFFFF
Modified data value (hex) = FFFFFE3F
 Decimal: 4294966847 Octal: 37777777077 Hex: FFFFFE3F Binary:
1111_1111_1111_1111_1111_1110_0011_1111
Enter command (or 'h' for help): n
Original data value (hex) = FFFFFE3F
Modified data value (hex) = 1C0
 Decimal:        448 Octal:         700 Hex:      1C0 Binary:
0000_0000_0000_0000_0000_0001_1100_0000
Enter command (or 'h' for help): o
Original data value (hex) = 1C0
Enter hex argument for Bitwise-OR: F000
Modified data value (hex) = F1C0
 Decimal:      61888 Octal:      170700 Hex:     F1C0 Binary:
0000_0000_0000_0000_1111_0001_1100_0000
Enter command (or 'h' for help): c
Original data value (hex) = F1C0
Enter digit number (count from the right, begin at zero): 2
Modified data value (hex) = 0
 Decimal:          0 Octal:           0 Hex:        0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): s
Original data value (hex) = 0
Enter argument for new data value (hex): 87654321
Modified data value (hex) = 87654321
 Decimal: 2271560481 Octal: 20731241441 Hex: 87654321 Binary:
1000_0111_0110_0101_0100_0011_0010_0001
Enter command (or 'h' for help): d
Original data value (hex) = 87654321
Enter digit number (count from the right, begin at zero): 3
Enter desired new value (hex): 9
Modified data value (hex) = 0
 Decimal:          0 Octal:           0 Hex:        0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): q
Are you sure that you want to exit the program? y
Exit the program.
```

Part 2: Modify the Program

Modify the PARTIALLY CODED functions for the ‘**c**’ command (**Clear Hex Digit**) and the ‘**d**’ command (**Define New Value for Hex Digit**).

Clear Hex Digit (‘c’ command)

Modify the **clearHexDigit()** function, so that it accepts a digit position (0-7) from the user and sets the bits for that digit to zero.

Define New Value for Hex Digit (‘d’ command)

Modify the **defineNewValueForHexDigit ()** function, so that it accepts a digit position (0-7) from the user and sets the bits for that digit to the new hex value entered by the user.

IMPORTANT restriction:

Do **NOT** make any changes to the code outside the two functions mentioned above:

clearHexDigit()

defineNewValueForHexDigit ()

The “starter code” for this Lab Exercise has been carefully written to assist you in understanding how bitwise operations work. Ill-advised changes to the “starter code” portion of this assignment have caused confusion to other students in the past.

Sample Interactive Session (solution)

The following table contains text that represents a possible execution session for the completed program. (In this example, what the user types is shown in **bold**. The purpose of this is to assist the reader of this document to distinguish text that the user entered from text that the program printed. When the program *actually* runs, text that the user types will appear in the same text format as the output.)

Part 2: Sample Interactive Session (solution)

```
Decimal:          0 Octal:           0 Hex:         0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): h
Supported commands:
     a    bitwise AND.
     c    Clear specific Hex digit. (Needs STUDENT modifications)
     d    Define new contents for specific Hex digit. (Needs STUDENT
modifications)
     h    Print this help text.
     l    (lower-case L) bitwise LEFT SHIFT
     n    Negate (invert) all bits.
     o    bitwise OR.
     r    bitwise RIGHT SHIFT.
     s    SET data value (hex).
     q    quit (end the program).
     v    SET or CLEAR verboseMode (debug aids).
     x    bitwise XOR.
```

Part 2: Sample Interactive Session (solution)

```
   demo   Demonstrate how to use verboseMode.

 Decimal:          0 Octal:           0 Hex:        0 Binary:
0000_0000_0000_0000_0000_0000_0000_0000
Enter command (or 'h' for help): s
Original data value (hex) = 0
Enter argument for new data value (hex): 12345678
Modified data value (hex) = 12345678
 Decimal:  305419896 Octal:  2215053170 Hex: 12345678 Binary:
0001_0010_0011_0100_0101_0110_0111_1000
Enter command (or 'h' for help): c
Original data value (hex) = 12345678
Enter digit number (count from the right, begin at zero): 5
Modified data value (hex) = 12045678
 Decimal:  302274168 Octal:  2201053170 Hex: 12045678 Binary:
0001_0010_0000_0100_0101_0110_0111_1000
Enter command (or 'h' for help): d
Original data value (hex) = 12045678
Enter digit number (count from the right, begin at zero): 2
Enter desired new value (hex): 9
Modified data value (hex) = 12045978
 Decimal:  302274936 Octal:  2201054570 Hex: 12045978 Binary:
0001_0010_0000_0100_0101_1001_0111_1000
Enter command (or 'h' for help): q
Are you sure that you want to exit the program? y
Exit the program.
```

Submit and Demonstrate the Working Program

Submit the source code file (`*.cpp`) for the working program to *Moodle* assignment for this Lab Exercise.
Demonstrate the working program to the instructor during class.
Be sure to save a copy of the source file (`*.cpp`) in a safe place for future reference.

Always remember: make small, incremental changes. Test each small change as you go.