Lab Practice Week 5
To be submitted as C Exercise 2 by Week 9

Internal Students: You need to show a working version of your solutions to program 10, 11 and 12. Your tutor will expect to see your submission on or before your lab class in week 9 as part of your C Exercise 2 assessment.

External Students: Please email your Program 10, 11 and 12 to your tutor. Your tutor will expect to receive them by the end of week 9.

Program 10

Write and test a C program that lists all the integer values between -56 and 77 in their internal computer representation which have bits 5 and bit 3 set (i.e. 1), and bit 2 not set (i.e. 0). Consider the 8 bit example below:

| Bit positions and requirements: 0 – not set, 1 – set, x – don’t care |
|------------------------|-----------------|
| 7 6 5 4 3 2 1 0        |
| x X 1 X 1 0 x X        |

For example,
- 77 in decimal equals to 1001101 – fails the requirement
- 40 in decimal equals to 101000 – satisfies the requirement

Hint: Requires the use of bit wise operation to check the bit positions. This program should be completed with as little number of lines of codes as possible.

Program 11

Write and test a C program that gets an integer from the user, and then multiplies that number by 32, and then divides that number by 16. Do not use multiplication, division or repeated addition.

Hint: Think about the logical position of a binary number. What happens if it is shifted left or right? How to program this in C? How about negative numbers? Can this be applied to Floating point numbers?
Program 12

Write and test a C program that reads a positive integer from the keyboard and then converts that number to one’s and two’s complement forms. Use only logical operators (and possibly addition).

Example:
- An 8bit integer: 1011 0110
  - One’s complement: 0100 1001
  - Two’s complement: 0100 1010

Assuming that you are using a 256 bit integer to represent a “key” in an encryption program, describe the algorithm in order to complete the same task. (Coding of this part of the program is optional.)

Note: While you may develop your program in any platform or compiler, all programs are expected to be able to run in the Cygwin environment. Even you may not be able to produce the complete program in the lab session, you MUST record and show your work to your tutor, otherwise no marks will be given. You should be able to give the following information:

1. An overall design of the program (How does the program work?)
2. Algorithm of the program (How do you process the data?)
3. Code and comments (What have you developed?)
4. Results (What are the outputs from the program?)
5. Testing (How did you test it?)
6. Discussion (Does it work? If no, what will you do next? If yes, how can it be improved?)