**PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**LAB WORK SHEET - 5**

**Class : M.Sc (SS/TCS) Semester: 1 Course : C Programming Lab**

**Bitwise Operators**

1. Write a program to read two numbers and print the result of all bitwise operations.

2. Write a program that right shifts an integer variable 4 bits. The program should print the integer in bits before and after the shift operation. (print all 32 bits use bitwise)

3. Left shifting an unsigned int by 1 bit is equivalent to multiplying the value by 2. Write a program that takes two integers number and pow and calculates number \* 2pow

4. Write a program that inverts the bits of an unsigned char X and stores answer in Y. Your answer should print out the result in binary form (although input can be in decimal form).

5. Write a program that rotates (**NOT shifts**) to the right by n bit positions the bits of an unsigned char x.ie no bits are lost in this process.

6. Write a program to do the following

Get the rightmost bit of any input

Get the 3 rightmost bits of any input

Get the leftmost bit of any input

Get the 3 leftmost bits of any input

Remove rightmost bit of any input

Remove rightmost 3 bits of any input

Remove leftmost bit of any input

Remove 3 leftmost bits of any input

Get 4 rightmost bits of any input, and remove the last bit

Remove first bit of any input, and add it to the right

7. Write a program to find the biggest among two numbers without using any control structures.

8. Write a program to swap values two integer variables using bitwise operators.

9. Examine the C program below, and analyse the value displayed by the program for each output operation.

#include <stdio.h>

int main()

{

signed int AAA = 0x7d0000f2, BBB = 0xb500004e;

unsigned int CCC = 0x7d0000f2, DDD = 0xb500004e;

printf( "\nUnary Operations\n\n" );

printf( "%08x\n", -AAA ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", ̃AAA ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", -BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", ̃BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "\nBinary Bitwise Operations (signed)\n\n" );

printf( "%08x\n", AAA & BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA ^ BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA | BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA << 4 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB << 4 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA >> 12 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB >> 12 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "\nBinary Bitwise Operations (unsigned)\n\n" );

printf( "%08x\n", CCC & DDD ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC ^ DDD ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC | DDD ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC << 4 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD << 4 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC >> 12 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD >> 12 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "\nBinary Arithmetic Operations (signed)\n\n" );

printf( "%08x\n", AAA + BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA - BBB ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB - AAA ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA \* 16 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB \* 16 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA / 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB / 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", AAA % 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", BBB % 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "\nBinary Arithmetic Operations (unsigned)\n\n" );

printf( "%08x\n", CCC + DDD ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC - DDD ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD - CCC ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC \* 16 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD \* 16 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC / 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD / 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC % 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD % 4096 ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "\nMasking Operations\n\n" );

printf( "%08x\n", CCC & 0xffff ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", CCC | 0xffff ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD & 0xffff ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

printf( "%08x\n", DDD | 0xffff ); /\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*/

}

10. Write a C Program to find the Highest Bit Set for any given Integer.

11. Write a C Program to check if the given number is in power of 2 or not.

12. Write a C Program to use Bitwise Operations to Round(floor of) an Integer to next Lower Multiple of 2.

13. Write a C Program to swap the ith and jth bits for a 32-bit integer.

14. Write a C program to count the number of leading and training zeros in the given integer.