

APPLIED COMPUTER SCIENCE
ACS-2906-002
Computer Architecture and System Software
Winter 2025
Lab 2

Motivation

The goal of this laboratory is to reinforce Bit-Level Operations.

Rule for this Laboratory

- Assumptions

Integers are represented in two's-complement form.

In Java, the operator `>>` performs an arithmetic right shift operation, whereas `>>>` performs a logic right shift.

- Forbidden

Conditionals (if or `?:`), loops, switch statements, function calls, and macro invocations.

Divisions, modulus, and multiplication

Relative comparison operators (`<`, `>`, `<=` and `>=`).

- Allowed operations

All bit-level and logic operations.

Left and right shifts, but only with shift amounts between 0 and $w - 1$,

Addition and subtraction

Equality (`==`) and inequality (`!=`) tests.

Integer constants `INT_MIN` and `INT_MAX`

Casting between data types `int` and unsigned, either explicitly or implicitly.

Standard input & output.

You can use any method to convert integer to binary representation.

Questions

1. (4 points) Write a Java program to swap all bits in odd positions with bits in even positions. The following is a sample output: User enters 682, which is 10 1010 1010 in binary and the output is 01 0101 0101.

You should swap the bit at the i_{th} position with the bit at the $i_{th}-1$ position, for $i > 0$ and $i \leq w - 1$, where w is the number of bits. So, you swap the bit at position 0 with the bit at position 1, the bit at position 2 with the bit at position 3, and so on.

input	1	0	1	0	1	0	1	0	1	0
bit position	9	8	7	6	5	4	3	2	1	0
output	0	1	0	1	0	1	0	1	0	1

Examples:

Enter a Number: 4

User Entered: 0100

Bits Swapped: 8 (1000)

Enter a Number: 1927

User Entered: 011110000111

Bits Swapped: 2891 (101101001011)

Enter a Number: -2147483648

User Entered: 10000000000000000000000000000000

Bits Swapped: 1073741824 (01000000000000000000000000000000)

Hint:

- Use a mask and bit-wise logic operation to extract the bits at odd positions.
- Use a mask and bit-wise logic operation to extract the bits at even positions.
- Shift and combine the bit vectors (think to which direction each bit vector should be shifted, and how to combine those 2 vectors)

Note: Leading zeros may not be displayed when treating results as integers. For example, treating 01 0101 0101 as an integer, will not produce a leading zero, i.e. the output will be 101010101 which is **acceptable**. You can use the Java function `Integer.toBinaryString(input)` to visualize the binary form of a number.

2. (3 points) Write a Java program, without using any if/else statements, that return 1 when a number is positive.

$$X(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}.$$

Hint: Which is the bit that indicates the sign in a number? Think about how to place that bit in the least significant position. You also need logic bit-wise operations to produce the desired output (1 for positive numbers).

3. (3 points) Write a java program that swaps values between two variables without using a third variable. For example:

Enter value for x: 20

Enter value for y: 30

Value of x: 30

Value of y: 20

Hint: Review the practice exercise 2.10 at page 90 in the text book (3rd. edition)

Evaluation:

- You **must** comment your code to explain that you understand underlying functionality to receive full marks.

Submission instructions

Include your name and student number in all files. Zip your file or files into one report named *StudentNumber_Lab2.zip*, and submit it through Nexus. **Students that do not follow these instructions will lose 2 marks. Late submissions will not be accepted. NO EXCEPTIONS.**