**Name**

**Advanced Programming in Java**

**Lab Exercise 10/17/2024**

**Lesson 28…..Bitwise Operators**

1. What is the bitwise operator for **AND**?

2. What is the *boolean* operator for **AND**?

3. What is the bitwise operator for **OR**?

4. What is the *boolean* operator for **OR**?

5. What is the bitwise operator for exclusive-**OR**?

6. What is the bitwise operator for **NOT**?

7. What is the *boolean* operator for **NOT**?

Use the following code to tell what’s printed in problems 8 – 18. (The code in some problems may not compile. If that’s the case then state, “Won’t compile.”)

int j = 79, k = 82, p = 112, q = 99;

8. System.out.println( (137) | q );

9. System.out.println( (137) & (121) );

10. System.out.println( (137) && (0x3A) );

11. System.out.println( (137) ^ (121) );

12. System.out.println( (p) | (j) );

13. System.out.println( ~ 465 );

14. System.out.println( j ^ (0x4B) );

15. System.out.println( (j) & (k) );

16. System.out.println( p || j );

17. System.out.println( p ^ q );

18. System.out.println( ~ (-k) );

19. What does msb stand for?

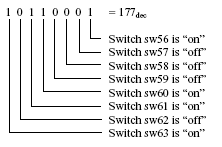
20. What do you get if you bitwise-exclusive-**OR** two 1’s?

21. What can be said about an integer if its most significant bit is 1?

22. An integer’s msb is 1. If this integer is multiplied by –27 what will be the resulting sign?

**Project… Masking Telemetry Data**

In a telemetry system in a spacecraft each bit position within the first eight bits of an integer sent to ground control have meaning with regard to the status of certain switches onboard the spacecraft. Assuming a 1 indicates the switch is **on** and a 0 indicates that it’s **off**, these meanings are illustrated below:



The numbers sent to ground control never exceeds 255 (first eight bits all set to 1).

Suppose we wish to look at the third bit from the left. Use a mask as illustrated below to

bitwise **AND** with the original number in order to look **only** at the third bit.



Notice this scheme of bitwise **AND-ing** a mask value of 32 (25, since the third bit position from the left has a positional value of 25) yields a value in the third bit position exactly equal to the third bit position of the original number. All other bit positions are guaranteed to be 0’s. Thus, this result of bitwise **AND-ing** can be tested to see if its entire value is 0. If it’s greater than 0, this means that the bit in the tested position was a 1.

Write a program that will input the following data file, *Switches.in*, containing decimal

numbers that represents successive switch information telemetry. Print the status of all eight switches.

**Switches.in**

22

194

203

97

Your output should be as shown on the next page:

Switch status for data value 22:

Switch sw56 is "off"

Switch sw57 is "on"

Switch sw58 is "on"

Switch sw59 is "off"

Switch sw60 is "on"

Switch sw61 is "off"

Switch sw62 is "off"

Switch sw63 is "off"

Switch status for data value 194:

Switch sw56 is "off"

Switch sw57 is "on"

Switch sw58 is "off"

Switch sw59 is "off"

Switch sw60 is "off"

Switch sw61 is "off"

Switch sw62 is "on"

Switch sw63 is "on"

Switch status for data value 203:

Switch sw56 is "on"

Switch sw57 is "on"

Switch sw58 is "off"

Switch sw59 is "on"

Switch sw60 is "off"

Switch sw61 is "off"

Switch sw62 is "on"

Switch sw63 is "on"

Switch status for data value 97:

Switch sw56 is "on"

Switch sw57 is "off"

Switch sw58 is "off"

Switch sw59 is "off"

Switch sw60 is "off"

Switch sw61 is "on"

Switch sw62 is "on"

Switch sw63 is "off"

**Project…Necklace**

An interesting problem in number theory is sometimes called the “necklace problem”. This problem begins with two single-digit numbers. The next number is obtained by adding the first two numbers together and saving only the unit’s digit. This process is repeated until the “necklace” closes by returning to the original two numbers. For example, if the starting two numbers are 1 and 8, twelve steps are required to close the necklace: 1 8 9 7 6 3 9 2 1 3 4 7 1 8.

Create a Necklace application that prompts the user for two single-digit numbers and then displays the sequence and the number of steps taken. The application output should look similar to:

Enter the first starting number: 1

Enter the second starting number: 8

1 8 9 7 6 3 9 2 1 3 4 7 1 8

Number of steps to close the necklace: 12