**1) Logical shift**

A logical shift is a bitwise operation that shifts all the bits of its operand. It does not preserve a number's sign bit or distinguish a number's exponent from its mantissa; every bit in the operand is simply moved a given number of bit positions, and the vacant bit-positions are filled in, usually with zeros.

**2) Arithmetic shift**

An arithmetic shift is a bitwise operation that shifts all of the bits of its operand; every bit in the operand is simply moved a given number of bit positions, and the vacant bit-positions are filled in. Instead of being filled with all 0s, as in logical shift, when shifting to the right, the leftmost bit (usually the sign bit in signed integer representations) is replicated to fill in all the vacant positions.

3) Rotate (hint: search for assembly rotate or bit rotate)

Rotate instructions are similar to shift instructions, ecept that rotate instructions are circular, with the bits shifted out one end returning on the other end. Rotates can be to the left or right. Rotates can also employ an extend bit for multi-precision rotates.

What resources did you use to find these answers?

Wikipedia and <http://www.osdata.com/topic/language/asm/shiftrot.htm#shiftandrotate>