HW2.2. Bitwise Mathematics (Randomized)

For the following questions assume that all values are 8-bit unsigned values.

Evaluate the following, and submit your answer as an 8-bit binary value. Don't forget to include the "0b"!

Q1.1:

0b10011111 ^ 0b11010100 | 0b01001011 ? ✓ 100%

Q1.2:

0b11100011 & 0b10111010 | 0b10100010 ? ✓ 100%

Evaluate the following, and submit your answer as a decimal integer. Note that >> is a right shift operator and << is a left shift operator.

Q2.1: 80>>3 | 10 ? ✓ 100%

One common application of bitwise operations is in state variables, where a number of related boolean variables are encoded as an integer. This is often more efficient than an array of bools, since it allows multiple bools to change in a single operation.

Integers are stored in computer memory as binary digits. Thus, you can think of integers as an array of 0s and 1s (but not explicitly declared as arrays).

Let `x` be an 8-bit unsigned number. Fill in the decimal integer in the following expressions such that the specified effect is achieved. We consider the "first bit" to be the least significant bit, and the "eighth bit" to be the most significant bit.

Note that for these questions **only the specified bits should be changed**. That is, the rest of the bits should remain unchanged after the bitwise operation.

Hint: Note that `x &= y` *really means x = x & y. You are looking for y.*

Q3.1: We want to turn on the third and fifth bits. *Hint: Write out the table for the OR operation*

x |= | 20 ? ✓ 100%

Q3.2: We want to turn off the fourth bit. *Hint: Write out the table for the AND operation*

x &= | 247 ? ✓ 100%

Q3.3: We want to flip the seventh bit. *Hint: Write out the table for the XOR operation*

x ^= | 64 ? ✓ 100%

Try a new variant

Correct answer

Evaluate the following, and submit your answer as an 8-bit binary value. Don't forget to include the "0b"!

Q1.1: 0b10011111 ^ 0b11010100 `0b01001011`

Q1.2: 0b11100011 & 0b10111010 `0b10100010`

Evaluate the following, and submit your answer as a decimal integer. Note that >> is a right shift operator and << is a left shift operator.

Q2.1: 80>>3 `10`

Homework 2

Assessment overview

Total points:	10/100
Score:	10%

Question

Value:	5
History:	5
Awarded points:	5/5

Report an error in this question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

One common application of bitwise operations is in state variables, where a number of related boolean variables are encoded as an integer. This is often more efficient than an array of bools, since it allows multiple bools to change in a single operation.

Integers are stored in computer memory as binary digits. Thus, you can think of integers as an array of 0s and 1s (but not explicitly declared as arrays).

Let x be an 8-bit unsigned number. Fill in the decimal integer in the following expressions such that the specified effect is achieved. We consider the "first bit" to be the least significant bit, and the "eighth bit" to be the most significant bit.

Note that for these questions **only the specified bits should be changed**. That is, the rest of the bits should remain unchanged after the bitwise operation.

Hint: Note that x &= y really means x = x & y. You are looking for y.

Q3.1: We want to turn on the third and fifth bits. *Hint: Write out the table for the OR operation*
x |= 20

Q3.2: We want to turn off the fourth bit. *Hint: Write out the table for the AND operation*
x &= 247

Q3.3: We want to flip the seventh bit. *Hint: Write out the table for the XOR operation*
x ^= 64

Submitted answer 2 correct: 100%
Submitted at 2022-09-03 08:24:37 (PDT)

hide

Evaluate the following, and submit your answer as an 8-bit binary value. Don't forget to include the "0b"!

Q1.1: 0b10011111 ^ 0b11010100 `0b01001011` ✓ 100%

Q1.2: 0b11100011 & 0b10111010 `0b10100010` ✓ 100%

Evaluate the following, and submit your answer as a decimal integer. Note that >> is a right shift operator and << is a left shift operator.

Q2.1: 80>>3 10 ✓ 100%

One common application of bitwise operations is in state variables, where a number of related boolean variables are encoded as an integer. This is often more efficient than an array of bools, since it allows multiple bools to change in a single operation.

Integers are stored in computer memory as binary digits. Thus, you can think of integers as an array of 0s and 1s (but not explicitly declared as arrays).

Let x be an 8-bit unsigned number. Fill in the decimal integer in the following expressions such that the specified effect is achieved. We consider the "first bit" to be the least significant bit, and the "eighth bit" to be the most significant bit.

Note that for these questions **only the specified bits should be changed**. That is, the rest of the bits should remain unchanged after the bitwise operation.

Hint: Note that x &= y really means x = x & y. You are looking for y.

Q3.1: We want to turn on the third and fifth bits. *Hint: Write out the table for the OR operation*
x |= 20 ✓ 100%

Q3.2: We want to turn off the fourth bit. *Hint: Write out the table for the AND operation*
x &= 247 ✓ 100%

Q3.3: We want to flip the seventh bit. *Hint: Write out the table for the XOR operation*
x ^= 64 ✓ 100%

Submitted answer 1 **partially correct: 83%**

Submitted at 2022-09-03 08:23:58 (PDT)

show