HW2.1. Bitwise Operations

C provides bitwise commands for AND(&), OR(|), XOR(^), and NOT(~).

Let x be a single-bit input (i.e. can be either 0 or 1). Fill in the following blanks with either `0`, `1`, `x`, or `~x`. Putting x means that `x` is the output (no change), and `~x` means that the opposite of `x` is the output.

Tip: Review AND, OR, XOR logic. What is the corresponding output for different input combinations using these logic?

Q1.1: $x\&1 =$ x ? ✓ 100%

Q1.2: $x\&0 =$ 0 ? ✓ 100%

Q1.3: $x \mid 1 =$ 1 ? ✓ 100%

Q1.4: $x \mid 0 =$ x ? ✓ 100%

Q1.5: $x\hat{}1 =$ ~x ? ✓ 100%

Q1.6: $x\hat{}0 =$ x ? ✓ 100%

Based on your responses above, which operation would be useful for:

Q2.1: turning bits ON (i.e. setting bits to 1)?

○ (a) &

● (b) | ✓

○ (c) ^

✓ 100%

Q2.2: turning bits OFF (i.e. setting bits to 0)?

● (a) & ✓

○ (b) |

○ (c) ^

✓ 100%

Q2.3: flipping bits (i.e. changing bits from 0 to 1 or from 1 to 0)?

○ (a) &

○ (b) |

● (c) ^ ✓

✓ 100%

Try a new variant

Homework 2

Assessment overview

Total points: 5/100

Score: 5%

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

Correct answer

C provides bitwise commands for AND(&), OR(|), XOR(^), and NOT(~).

Let x be a single-bit input (i.e. can be either 0 or 1). Fill in the following blanks with either `0`, `1`, `x`, or `~x`. Putting x means that `x` is the output (no change), and `~x` means that the opposite of `x` is the output.

Tip: Review AND, OR, XOR logic. What is the corresponding output for different input combinations using these logic?

Q1.1: $x \& 1 =$ `x`

Q1.2: $x \& 0 =$ `0`

Q1.3: $x \mid 1 =$ `1`

Q1.4: $x \mid 0 =$ `x`

Q1.5: $x \hat{} 1 =$ `~x`

Q1.6: $x \hat{} 0 =$ `x`

Based on your responses above, which operation would be useful for:

Q2.1: turning bits ON (i.e. setting bits to 1)?
(b) $\mid$

Q2.2: turning bits OFF (i.e. setting bits to 0)?
(a) $\&$

Q2.3: flipping bits (i.e. changing bits from 0 to 1 or from 1 to 0)?
(c) ^

Submitted answer 2 **correct: 100%**
Submitted at 2022-09-03 08:17:10 (PDT)

ℹ hide ⌃

C provides bitwise commands for AND(&), OR($\mid$), XOR(^), and NOT(`~`).

Let x be a single-bit input (i.e. can be either 0 or 1). Fill in the following blanks with either `0`, `1`, `x`, or `~x`. Putting x means that `x` is the output (no change), and `~x` means that the opposite of `x` is the output.

Tip: Review AND, OR, XOR logic. What is the corresponding output for different input combinations using these logic?

Q1.1: $x \& 1 =$ `x` ✓ 100%

Q1.2: $x \& 0 =$ `0` ✓ 100%

Q1.3: $x \mid 1 =$ `1` ✓ 100%

Q1.4: $x \mid 0 =$ `x` ✓ 100%

Q1.5: $x \hat{} 1 =$ `~x` ✓ 100%

Q1.6: $x \hat{} 0 =$ `x` ✓ 100%

Based on your responses above, which operation would be useful for:

Q2.1: turning bits ON (i.e. setting bits to 1)?
(b) $\mid$ ✓ 100%

Q2.2: turning bits OFF (i.e. setting bits to 0)?
(a) $\&$ ✓ 100%

Q2.3: flipping bits (i.e. changing bits from 0 to 1 or from 1 to 0)?
(c) ^ ✓ 100%

Submitted answer 1 **invalid, not gradable**
Submitted at 2022-09-03 08:15:46 (PDT)

ℹ show ⌄