CS161 Spring 2026

Introduction to Computer Security

Discussion 0

Q1 *61C Review* (11 points)

Being comfortable manipulating the various number representations covered in 61C will help you succeed in the memory safety unit.

Q1.1 (1 point) What is the hexadecimal value of the decimal number 18?

`0x`

Q1.2 (1 point) What is the value of `0x8339e833 + 0x20` in hexadecimal form?

`0x`

Q1.3 (1 point) What is the value of `0x550ecdf2 + decimal 16` in hexadecimal form?

`0x`

Q1.4 (1 point) What is the largest unsigned 32-bit integer? What is the result of adding 1 to that number?

`max: 0x`	`max + 1: 0x`

Q1.5 (1 point) What is the largest signed 32-bit integer? What is the result of adding 1 to that number?

`max: 0x`	`max + 1: 0x`

Q1.6 (1 point) If you interpret an n-bit two's complement number as an unsigned number, would the negative numbers be smaller or larger than positive numbers?

○ Smaller ○ Larger

Q1.7 (1 point) How many bytes are needed to represent `char[16]`?

`bytes`

Q1.8 (1 point) How many bytes are needed to represent `int[8]`?

`bytes`

This content is protected and may not be shared, uploaded, or distributed.

(Question 1 continued...)

Q1.9 (1 point) In a little-endian 32-bit system, how would you represent the pointer `0xDEADBEEF`?

0x	0x	0x	0x

Q1.10 (1 point) In a little-endian 64-bit system, how would you represent the pointer `0xDEADBEEF`?

0x	0x	0x	0x	0x	0x	0x	0x

Q1.11 (1 point) In a little-endian 32-bit system, how would you represent the char array “ABCDEFGH”?

Recall that our stack representation has addresses increase from left-to-right and bottom-to-top.

This content is protected and may not be shared, uploaded, or distributed.