

**Question 1** *61C Review*

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

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

**Solution:** 0x12

Q1.2 What is the value of  $0x8339e833 + 0x20$  in hexadecimal form?

**Solution:** 0x8339e853

Q1.3 What is the value of  $0x550ecdf2 + \text{decimal } 16$  in hexadecimal form?

**Solution:** 0x550ece02

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

**Solution:** The largest 32-bit unsigned integer is 0xffffffff, and you will get 0x00000000 if you add 1 to it (unsigned overflow).

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

**Solution:** The largest 32-bit signed integer is 0x7fffffff, and you will get 0x80000000 if you add 1 to it, which will be  $-2^{31}$  in decimal (signed overflow).

Q1.6 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?

**Solution:** Negative numbers would be larger than positive numbers if interpreted as unsigned since their most significant bits are set.

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

**Solution:** A character array with 16 elements in it needs 16 bytes to represent because each character is 1 byte.

Q1.8 How many bytes are needed to represent `int[8]`?

**Solution:** An integer array with 8 elements in it needs 32 bytes to represent because each integer is 4 bytes.

Q1.9 For the following subparts, assume each block is 1 byte, and addresses increase from left-to-right and bottom-to-top.

In a little-endian 32-bit system, how would you represent the pointer `0xDEADBEEF`?

**Solution:**

0xEF	0xBE	0xAD	0xDE
------	------	------	------

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

**Solution:**

0xEF	0xBE	0xAD	0xDE	0	0	0	0
------	------	------	------	---	---	---	---

Q1.11 In a little-endian 32-bit system, how would you represent the char array "ABCDEFGH"?


**Solution:**

E	F	G	H
A	B	C	D