# 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?

`0x12`

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

`0x8339e853`

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

`0x550ece02`

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

`max: 0xffffffff` | `max + 1: 0x00000000`

**Solution:** This is a result of unsigned overflow.

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

`max: 0x7fffffff` | `max + 1: 0x80000000`

**Solution:** This is a result of signed overflow. `0x7fffffff` is $2^{31} - 1$ in decimal and `0x80000000` is $-2^{31}$ in decimal.

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

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

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

| `16 bytes` |
|---|

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

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

| `32 bytes` |
|---|

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

| `0xEF` | `0xBE` | `0xAD` | `0xDE` |
|---|---|---|---|

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

| `0xEF` | `0xBE` | `0xAD` | `0xDE` | `0x00` | `0x00` | `0x00` | `0x00` |
|---|---|---|---|---|---|---|---|

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.

| E | F | G | H |
|---|---|---|---|
| A | B | C | D |