Computer Memory

The bit is the smallest unit of information stored in a computer. Each bit can be in one of two possible states: 0 or 1, representing off or on or true or false. Individual bits are grouped together into groups of 8 to create bytes, and the byte is how numbers are actually stored. (Everything is stored as a number, even strings) The value of the byte is determined from the state of the bits.

Each position in the byte represents a value:

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

The top row is the decimal value and the bottom row is the power of 2 equivalent for the decimal value.

If the bit at a position is set to 1, then that decimal value is included in calculating the byte value. For example,

128	64	32	16	8	4	2	1
0	0	0	1	1	0	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

= 2 + 8 + 16 = 26.

Or

$= 2^1 + 2^3 + 2^4 = 26$

Binary and hexadecimal
The 00011010… is a binary digit, and the conversion to 26 was binary to decimal. But, addresses and values are generally reported as hexadecimal, which is base 16. Digits 0 – F represent values from 0 – 15.

Digits 0 – 9 in decimal and hexadecimal are the same. However, digits 10 – 15 are A – F in hex.
A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

What is the maximum decimal value for one byte? The maximum value has all positions = 1, which is 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

What is the maximum value of the first four bits in a byte?

128	64	32	16	8	4	2	1
0	0	0	0	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

= 8 + 4 + 2 + 1 = 15.

The 15 is also the maximum hex value of F, since F = 15. If we use a hex representation of a byte, we don't use decimal values of a position, we split the byte into two groups of four bits and each group gets a hex value.

Instead of a byte having a maximum value of 255, it has a maximum hex value of FF.

128	64	32	16	8	4	2	1
0	0	0	0	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
F				F			

Multiple bytes of information
Clearly, we need to represent numbers larger than 255. For example, how do we represent:

intI = 1000

1000 is greater than 255. We need more than one byte. Let's say we have two bytes for storage. We continue with the powers of 2 to represent larger values using 16 bits of information, each with a value of 0 or 1.

32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3			

You use the same process to determine the value of the 2-bytes, summing the values that are set to 1. For example,

32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	0
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}			

= 32768 + 8192 + 2048 + 1024 + 128 + 32 + 8 + 2 = 44202

Once we have more than one byte, it becomes much easier to represent the value using hex because you only have to add to 15.

ACAA

The computer only sees 0 and 1, so the representation here is just for human readability benefit.

The maximum value of 2 bytes is the sum of all positions, 65535. You can also get this by subtracting 1 from the next power of two. The maximum bit value is 2^{15}. Therefore, the next bit value, if there were one, would be 2^{16}. The maximum value of two bytes is $2^{16} – 1$. Look at a smaller example to prove to yourself that it's true. The maximum value of one byte is 255, which is $2^8 – 1$. The leftmost (yes, I know endianness matters) position in one byte represents 2^7. The first position (right-most) in the second byte is 2^8.

Computer Memory
Bits make up the values stored, and data types designate how many bits each variable can have. Some common data types are:

int: 4 bytes
char: 1 byte
float: 4 bytes
long: 8 bytes
double: 8 bytes

The char data type will treat the number stored as a letter when it is used. For example, char 'A' is stored in memory as 65. The ascii table defines all char codes.

Variables are storage locations in memory. When we declare a variable, such as

int I;

That creates a label called I that is affiliated with some location in memory, and that location will have a value. Each memory location has an address, and the location stores one byte of information. For example,

Address	Value
0xFF06	
0xFF05	
0xFF04	
0xFF03	
0xFF02	AC
0xFF01	AA
0xFF00	

There are two bytes of information stored at locations 0xFF01 and 0xFF02. (The 0x designates that the value is in hex.)