ICS 3U/3C **Counting Systems and Machine Code (21)** Name:

As humans we use base ten, or decimal as our number system. However, any other number may be used as the base of a number system. The different bases simply determine the limit of a single number place.

For Base 10 (decimal), the limit is 10 digits:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…

For base 1, the limit is 1 digit:

1, 11, 111, 1111, 11111, 111111, 1111111, 11111111, 111111111, 1111111111…

So essentially base 1 is tallying.

Activity 1 - Convert these numbers from decimal to base 1:

a) 1

b) 5

c) 25

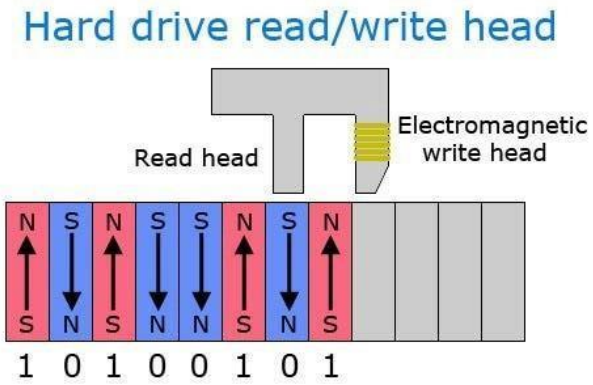
Activity 2 - Convert the following from base 1 to decimal:

a) 1

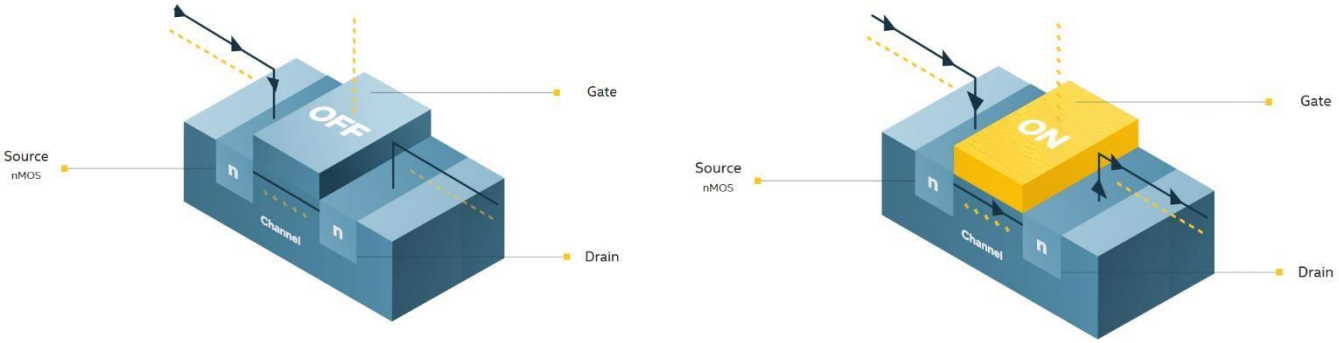
b) 11111

c) 11111111111111111111111111111111111111111111111111111111111111111111111111111
1111111

Computers use different based number systems for a variety of reasons. **Binary counting** (base 2) is the basis of all computing. **Machine code** is the most rudimentary form of code, which is directly read by the computer. It is written in binary, as a series of 0's and 1's. The 1's and 0's of binary are used to dictate the state of many 2 state devices that make up our computers, most notably, bits and transistors:



Bits are the storage units of computers. In a hard drive, the magnetized poles on the different bits correspond to binary code depending on their direction. The "read/write head" can store data by magnetizing bits, or take data by reading the magnetic poles.



The **transistors** direct the flow of current through a circuit depending on the state of it's gate. Binary code most often determines the state of a transistor. In this example, a 0 would represent OFF, and no current would flow. A 1 would represent ON, and current would flow through the transistor.

Different transistors may direct flow in multiple directions rather than simply impeding it as well. These transistors are the "switches" that determine all of the logic in a computer. Together they form complex circuits that can compute various operations from basic arithmetic to video games.

Activity 3 - Convert the following from binary to decimal:

a) 10

b) 110

c) 01101

d) 111111

Binary is efficient because it can easily be translated into functions on our current computers, as they use 2 state components. However, it is not the most efficient for data storage. Binary code

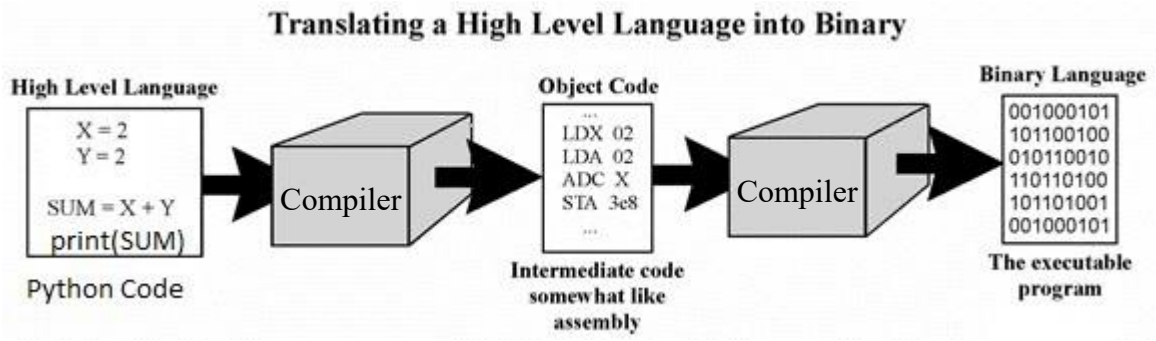
stores 2^n bits of data where n is the length of the string. Comparatively, **Hexadecimal** (base 16) stores 16^n bits of data. At large scales, it's much more efficient to store code as larger systems such as hexadecimal, then convert back to binary later.

When we look at different computing systems, they will often say how many bits they run on. That simply describes how many bits it can compute at once. For a 16-bit system, it will compute 16 digits of binary together at once. Here the machine code will be stored in 16-digit groupings and look something like this:

0101110101010110 0101110100001110 1101110101011111 1101110101011111
0101110101011111 0101110101011110 0101110101011110 0101110101011110
1101110101011110 1101110101011110 0101000101000110 0101111101011111

Not very user friendly or understandable. This is one of the reasons why we use **high-level programming languages** such as python. These languages are built to be more understandable and intuitive, making them easier to use. Additionally, they are much quicker to use rather than coding individual bit functions. These languages **compile** (translate) the understandable code through various stages, eventually into binary machine code.

The compilation process from python to machine code is a bit more complicated than just a single step, and will actually look something like this:



Activity 4 – Convert the following from decimal to hexadecimal:

For the digits beyond 9 in hexadecimal, we use letters: A B C D E F.

Counting from 1 to 16 looks like: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

a) 16

b) 21

c) 256

Activity 5 – Convert the following 16 binary groupings into hexadecimal.

a) 1111000000010100

b) 0001101101100111

c) 1111111111111111

d) 0000000000000000

Activity 6 – Convert the following hexadecimal groupings into 16 bit binary strings. a)

0000

b) ABCD

c) F10E

d) 1C34

Note: Binary is both a counting system AND the machine language of computers. In this way, we are able to convert it into smaller forms of storage such as hexadecimal (using the counting system), but it must be converted back into binary to be interpreted by a computer (machine code).