- difference between traditional variables and pointers
    - int x = 5
        - Creates an integer variable that stores 5
    - int* p = &x
        - creates a pointer that stores the memory address of int x
- pointer declaration
    - declaring a pointer
        - int* p; int* p = NULL
            - creates an empty or null pointer
    - assigning a pointer
        - int* p = &x
            - creates a new pointer and assigns memory address of x
- Difference between different pointer type declarations
    - Int* x
    - Char* y
    - double z
    - differences are the sizes appointed to each pointer type
- Reference and deference operator
    - &
        - gets the memory address
        - used when assigning address to a pointer or when passing pointer to a function
    - *
        - gets value at the memory address
        - used by function to get data from pointer when parameters are passed by pointer
- Pointers in functions with multiple outputs
    - parameters can be passed as pointers and referenced so the original value is changed
- arrays as pointers
    - The array variable stores the memory address of the first address of the array
    - incrementing a pointer gets the next memory address
    - This can be used to go through the elements of a list by incrementing the pointer to get the address of the next element of the array
- Argcc and argv in int main(int argc, char* argv[])
    - argc
        - number of command line arguments
    - argv
        - array of strings for command line arguments
    - used when processing command line arguments
        - passing input, filenames
        -