assignment 1

Declared an integer and a pointer to it. Printed the address using both & and the pointer, then modified the value using the pointer. The pointer directly accesses and modifies the memory location of the variable.

assignment 2

Declared an array and used a pointer to traverse and modify values. Pointer arithmetic (*(p+i)) gives access to array elements just like arr[i].

assignment 3

Implemented a swap function using pointers. Passed variable addresses to swap values. Functions can manipulate values outside their scope by using pointers, enabling pass-by-reference behavior.

assignment 4

Declared a variable, a pointer to it, and a double pointer. Printed the value via both levels of indirection. Double pointers allow multiple levels of indirection, useful in dynamic memory (e.g., malloc) and data structures.

assignment 5

Declared a string as a character array, traversed and printed using a pointer, and counted characters with pointer arithmetic. Strings in C are arrays of characters, and pointers enable flexible traversal and manipulation.