// ---------------------------- COMPILATION ------------------------------

// Preprocessing: Provides for the ability to include header files, macro expansions, conditional compilation, line control

// Compiling: Compiles the C file into assembly-level instructions that the computer can actually understand

// Assembling: Assembly is converted into binary so that the computer can actually understand it

// Linking: Includes library files in the machine code

// ------------------------------ POINTERS -------------------------------

int num = 5; // creates an int object with a value of 5

int \* pointer1 = num; // creates a pointer that points to object address

int \*\* pointer2 = &pointer1 // you can also have double pointers

// & means address and \* deferences a pointer when you wanna print it

printf(“%d”, \*pointer1); // prints 5

printf(“%p”, pointer1); // prints the address of num

// %d or %i is decimal, %p is pointer, %s is string for printf

// ------------------------------- STRINGS -------------------------------

// There are no actual strings in C, just arrays of // chars.

// so to make a string, you’d do

char str1[15] = “batman”;

char str2[10] = “robin”;

// you can also concat strings together with the strcat() function

strcat(str1, str2);

printf(“%s\n”, str1); // prints out batmanrobin

// you can also copy strings from 1 location to another.

strcpy(str1, str2);

printf(“%s\n”, str1); // prints out robin

// ------------------------------- ARRAYS --------------------------------

// arrays can be made much the same way as in Java, 1D and 2D ☺

int arr[10] = {1,2,3,4,5,6,7,8,9,10};

printf(“%d”, arr[3]); // prints 4

printf(“%d”, \*(arr + 3)) // also prints 4

int arr[10][10] = {{1,2,3}, {4,5,6}, {7,8,9}};

printf(“%d”, arr[1][2]) // prints 6

printf(“%d”, \*(\*(arr + 1) + 2)) // also prints 6

// ----------------------------- STRUCTS ---------------------------------

struct SvadK {   // Structure declaration  
  int myNum;           // Member (int variable)  
}; // End the structure with a semicolon

SvadK structure; // initializes a structure

structure.num = 5; // sets the value of num of initialized structure to 5

printf(“%d”, structure.num); // prints 5