1. The value printed when you print the value held by a pointer variable depends on the type of the pointer and the way it was initialized.
   1. If the pointer points to a basic data type such as an integer or a character, printing the value held by the pointer variable will typically display the value stored in the memory location pointed to by the pointer.
2. The return type of a comparison operator is typically a Boolean value. In C and C++, the Boolean data type is represented using the bool keyword and has two possible values: true and false.
   1. In C++, the comparison operators such as ==, !=, <, >, <=, and >= return a bool value of true or false depending on the outcome of the comparison.
   2. Comparison Operator is Bool type.
3. In C and C++, both structures and unions are used to group related data together. However, there are some important differences between the two:
   1. Memory allocation: In a structure, each member variable is allocated its own separate memory space. In contrast, in a union, all members share the same memory space. This means that a union can only hold one member value at a time.
   2. Size: The size of a structure is determined by the sum of the sizes of its member variables, while the size of a union is equal to the size of its largest member variable.
   3. Access to members: In a structure, each member variable can be accessed and modified independently. In a union, only one member can be accessed and modified at a time.
   4. Default initialization: In a structure, each member variable is initialized to a default value depending on its data type (e.g., 0 for integers, NULL for pointers, etc.). In a union, only the first member is initialized by default.
   5. Usage: Structures are typically used to group related data together that belong to the same entity, while unions are used to represent different views of the same data.
4. Pointer Arithmetic:

Main(){

Int A[5] = {5,8,9,6,10};

Int \*p,\*q;

P = &A[0];

Q = &A[3];

Size of int = 2bytes;

}

Pointer Increments

Main(){

P++;

}

Pointer Decrements

Main(){

p--;

}