Q1. What is the purpose of the main() function in a C program? Explain its significance.

In C programming, the main() function is extremely important because it is the point from where program execution actually begins. When we run a program, the operating system looks for the main() function as the entry point. Without this function, a C program will not execute. It can return an integer value to the operating system to indicate whether the program ended successfully or encountered an error. For example, returning 0 usually means the program has run successfully. In short, main() defines the beginning of the program and its completion.

Q2. Explain the difference between a variable declaration and a variable initialization in C.

In C, variable declaration and initialization are two related but different concepts. A declaration tells the compiler what type of data the variable will store and reserves the necessary memory space. For example, writing int x; only declares that x is an integer variable but does not give it a value. On the other hand, initialization means assigning an initial value to that variable at the time of declaration. For example, int x = 10; both declares the variable and also initializes it with the value 10. So, declaration is about creating the variable, while initialization is about giving it a starting value.

Q4. What are the different data types available in C? Provide examples of each data type.

The C programming language supports several basic data types that help programmers store different kinds of information. The most commonly used are: - int: used for storing whole numbers such as int age = 25; - float: used for storing decimal or fractional values such as float price = 99.50; - double: used when more precision is needed for decimal numbers such as double pi = 3.141592; - char: used for storing single characters such as char grade = 'A'; - void: represents nothing and is often used in functions that do not return a value. These data types allow us to handle numbers, characters, and other forms of information efficiently in C programs.

Q5. Explain the concept of type conversions in C. Provide examples of implicit and explicit type conversions.

Type conversion in C refers to changing the type of a variable from one data type to another. There are two types of conversions. The first is implicit conversion, also known as type promotion, which is done automatically by the compiler. For example, if we write int x = 5; float y = x + 2.5;, the integer x will be automatically converted to float before the calculation. The second type is explicit conversion, also known as type casting, which is done manually by the programmer. For instance, writing float y = (float)7 / 2; forces the integer 7 to be treated as a float, giving us a more accurate result. These conversions ensure that operations between different data types produce correct and expected results.

Q7. What is the role of the scanf() function in C? Provide an example of its usage.

The scanf() function in C is used to take input from the user. It allows us to enter values from the keyboard and store them into program variables. This function uses format specifiers like %d for integers, %f for floating point numbers, and %c for characters. For example, if we want to read an integer from the user, we can write: int age; scanf("%d", &age;); Here, the value entered by the user will be stored in the variable age. Therefore, scanf() plays a vital role in making programs interactive by letting users provide input at runtime.