Programming for Problem Solving

Q1. Type Casting

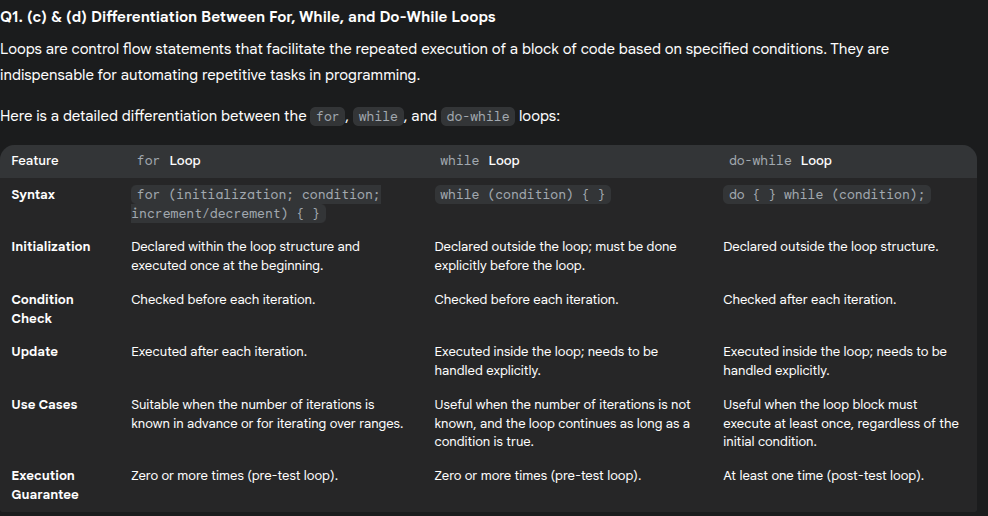
Ans. Type casting in C refers to the process of converting one data type to another desired one. This operation is fundamental, enabling variables of one data type to be utilized in operations or assignments that anticipate a different type, thereby ensuring compatibility and the desired level of precision. It allows for the modification of a variable's data type, tailoring it to meet specific program requirements.

Q2. Relational Operators

Ans. These operators compare two operands and return a boolean result (1 for true, 0 for false).

* == (equal to), > (greater than), < (less than), != (not equal to), >= (greater than or equal to), <= (less than or equal to).
* Example: 6 == 5 evaluates to 0 (false), while 6 > 5 evaluates to 1 (true).

Q3 & 4. Difference between for, while and do-whilel loops

Ans. 

Q4. Pseudocode

Ans. Pseudocode is a step-by-step description of an algorithm in a code-like structure using plain English text, designed to be easily read and understood by humans.

Q5.Different notations in a flowchart.

Ans. Common notations (symbols) used in flowcharts include:

* **Terminal (Oval/Rounded Rectangle):** Represents the start or end of a process.
* **Process (Rectangle):** Represents a step or action.
* **Decision (Diamond):** Represents a point where a decision is made, typically with two outcomes (e.g., Yes/No).
* **Input/Output (Parallelogram):** Represents data input or output operations.
* **Flowline (Arrow):** Connects symbols and indicates the direction of flow.

Q6. What are #include and #define directives?

Ans. The #include and #define are preprocessor directives in the C programming language. The #include directive is used to include the contents of a header file into the source code file. This allows the programmer to use functions, variables, and other declarations defined in the included header file.

On the other hand, the #define directive is used to define symbolic constants or macros. It replaces every occurrence of the defined identifier with the specified replacement text during the preprocessing phase. This can be used to create constants or to define macros that can perform operations or generate code.

Q7. Explain the various features of pointers. Write a C program to print address of a variable using pointers.

Ans. Pointers provide C programmers with significant power to directly manipulate memory. This enables the creation of highly optimized code and flexible data structures.

However, this power is accompanied by substantial risks.

Misusing pointers, such as dereferencing a null or invalid pointer, can lead to critical runtime errors like segmentation faults, memory leaks, or corrupted data.

This design philosophy in C emphasizes low-level control, but at the cost of increased programmer responsibility for ensuring memory safety.

The ability of pointers to link disparate memory locations is the foundational element for implementing dynamic data structures like linked lists, trees, and graphs. Unlike arrays, which store elements contiguously, these structures allow data nodes to be scattered across memory while maintaining logical connections through pointers.

Q8.