# Theory Question

27\_jan\_se NAME: Vrinda Mavadhiya

Q1Write an essay covering the history and evolution of C programming. Explain its importance and why it is still used today

:- C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.

It is a very popular language, despite being old.

The main reason for its popularity is because it is a fundamental language in the field of computer science.

C remains important because of its efficiency, low-level access to hardware, and portability, making it ideal for system-level programming, embedded systems, and performance-critical applications where speed and resource management are crucial. .

Q2]. Describe the steps to install a C compiler (e.g., GCC) and set up an Integrated Development Environment (IDE) like DevC++, VS Code, or CodeBlocks.

To install any language first of we need a source from where we can download the application. For languages like C, C++, DEVC++, CODEBLOCKS, this all are easily available to download from goggle and nowadays VSCODE is installed priorly in the system when purchased

We need to follow simple steps of installation and there is the application is done..!

Q3]. Explain the basic structure of a C program, including headers, main function, comments, data types, and variables. Provide examples.

A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files. All lines that start with **#**are processed by a preprocessor which is a program invoked by the compiler.

[main() function](https://www.geeksforgeeks.org/main-function-in-c/). It is the entry point of a C program and the execution typically begins with the first line of the main(). The empty brackets indicate that the main doesn’t take any parameter (See [this](https://www.geeksforgeeks.org/difference-int-main-int-mainvoid/)for more details). The int that was written before the main indicates the return type of main().

It is a [comment](https://www.geeksforgeeks.org/c-comments/). They are used for the documentation of the code and are ignored by the compiler.

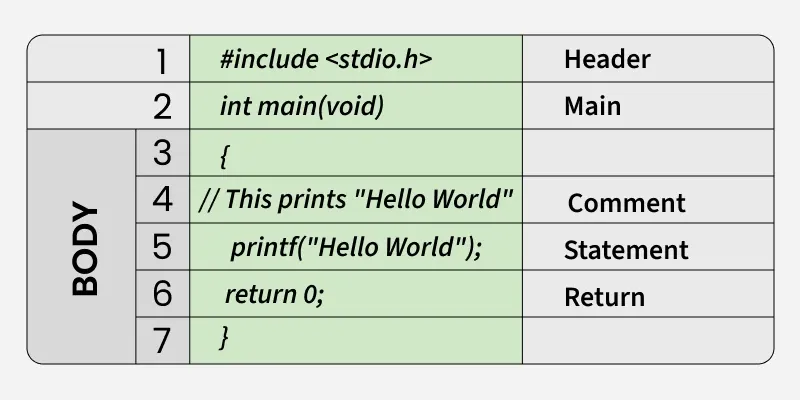
Each variable in C has an associated data type. It specifies the type of data that the variable can store like integer, character, floating, double, etc. Each data type requires different amounts of memory and has some specific operations which can be performed over it. **The following are some main primitive data types in C:**

**Table of Content**

* [Integer Data Type](https://www.geeksforgeeks.org/data-types-in-c/?ref=lbp#integer-data-type)
* [Character Data Type](https://www.geeksforgeeks.org/data-types-in-c/?ref=lbp#character-data-type)
* [Float Data Type](https://www.geeksforgeeks.org/data-types-in-c/?ref=lbp#float-data-type)
* [Double Data Type](https://www.geeksforgeeks.org/data-types-in-c/?ref=lbp#double-data-type)
* [Void Data Type](https://www.geeksforgeeks.org/data-types-in-c/?ref=lbp#void-data-type)

In C, **variable** is a name given to the memory location that helps us to store some form of data and retrieves it when required

THE STRUCTURE OF C PROGRAMMING LANGUAGE…



Q4]. Write notes explaining each type of operator in C: arithmetic, relational, logical, assignment, increment/decrement, bitwise, and conditional operators.

**Arithmetic Operator:** It is also called as 'Binary operators'. It is used to perform arithmetical operations. These operators operate on two operands.

**Relational operator**: is used when there is relation between 2 or more variables.

**Logical Operator:** They compare or evaluate logical and relational expressions.

**Assignment Operator:** It is used to assign a value to variable.

**Increment/Decrement Operator**

•It operates on a single operand. Therefore, this operator is called as 'unary operator.' It is used to increase or decrease the value of variable by 1.

**Conditional Operator**

•Conditional operator is also called as 'ternary operator.‘

•It is widely used to execute condition in true part or in false part. It operates on three operands.

•The logical or relational operator can be used to check conditions.

Q5]. Explain decision-making statements in C (if, else, nested if-else, switch). Provide examples of each.

**1.TheifStatement:**

•To conditionally execute statements, you can use the if or the if...else statement.

•The general form of the if statement isif(expr){s1;s2;....}

**2. The If else construct:**

•The if else is just an extension of the general format of if statement. If the result of the condition is true, then program statement 1 is executed, otherwise program statement 2 will be executed.if(condition)simpleorcompoundstatementelsesimpleorcompoundstatement.

**3. Nested if Statement:**

•The if statement may itself contain another if statement is known as nested if statement.

**4.The ELSE If Ladder**

•When a series of many conditions have to be checked we may use the ladder else if statement which takes the following general form. if (condition1) statement –1; else if (condition2) statement2; else if (condition3) statement3; else if (condition) statement n; else default statement; statement-x;

•This construct is known as if else construct or ladder. The conditions are evaluated from the top of the ladder to downwards.

5.**The Switch Statement:**

•Switch case statements are a substitute for long if statements that compare a variable to several integral values

•The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

•Switch is a control statement that allows a value to change control of execution.switch(switch\_expr){caseconstantexpr1:S1;S2;break;caseconstantexpr1:S3;S4;break;.....default:S5;S6;break;}

Q6]. Compare and contrast while loops, for loops, and do-while loops. Explain the scenarios in which each loop is most appropriate

**While Loop**: Condition is checked **before** the first iteration. If false initially, the loop won't run at all.

**For Loop**: Condition is checked **before** each iteration. Typically used when the number of iterations is known.

**Do-While Loop**: Condition is checked **after** each iteration. The loop always runs at least once.

**COMPARE AND CONTRAST**

**While Loop**: Use when you don't know how many times the loop should run, but you have a condition that will stop the loop. For example, reading user input until valid data is received.

**For Loop**: Use when you know how many times you want to repeat the loop, or when iterating through a collection. For example, iterating through a range of numbers or processing elements in an array.

**Do-While Loop**: Use when you need to execute the loop body at least once, even if the condition is false initially. For example, showing a menu to the user at least once and then repeating it based on user input.

Q7]. Explain the use of break, continue, and goto statements in C. Provide examples of each.

The continue statement in C is a jump statement used to skip the current iteration of a loop and continue with the next iteration. It is used inside loops (for, while, or do-while) along with the conditional statements to bypass the remaining statements in the current iteration and move on to next iteration.

#include <stdio.h>

int main() {

// Loop from 1 to 5

for (int i = 1; i <= 5; i++) {

// Skip 3 numbers

if (i == 3)

continue;

printf("%d ", i);

}

return 0;

}

The break in C is a loop control statement that breaks out of the loop when encountered. It can be used inside loops or switch statements to bring the control out of the block. The break statement can only break out of a single loop at a time.

#include <stdio.h>

int main() {

for (int i = 1; i <= 10; i++) {

// Exit the loop when i equals 5

if (i == 5) {

break;

}

printf("%d ", i);

}

return 0;

}

The goto statement in C allows the program to jump to some part of the code, giving you more control over its execution. While it can be useful in certain situations, like error handling or exiting complex loops, it’s generally not recommended because it can make the code harder to read and maintain.

#include <stdio.h>

int main()

{

int n = 0;

// If the number is zero, jump to

// jump\_here label

if (n == 0)

goto jump\_here;

// This will be skipped

printf("You entered: %d\n", n);

jump\_here:

printf("Exiting the program.\n");

return 0;

}

Q8].What are functions in C? Explain function declaration, definition, and how to

call a function. Provide examples.

A function in C is a set of statements that when called perform some specific tasks. It is the basic building block of a C program that provides modularity and code reusability. The programming statements of a function are enclosed within { } braces, having certain meanings and performing certain operations. They are also called subroutines or procedures in other languages.

**Function Declarations**

In a function declaration, we must provide the function name, its return type, and the number and type of its parameters. A function declaration tells the compiler that there is a function with the given name defined somewhere else in the program.

**Function Definition**

The function definition consists of actual statements which are executed when the function is called (i.e. when the program control comes to the function).

A C function is generally defined and declared in a single step because the function definition always starts with the function declaration so we do not need to declare it explicitly

**Function Call**

A function call is a statement that instructs the compiler to execute the function. We use the function name and parameters in the function call.

Q9].Explain the concept of arrays in C. Differentiate between one-dimensional and

multi-dimensional arrays with examples.

Array in C is one of the most used data structures in C programming. It is a simple and fast way of storing multiple values under a single name.

**1. One Dimensional Array in C**

The One-dimensional arrays, also known as 1-D arrays in C are those arrays that have only one dimension.

**2. multi-dimensional array in c**

can be defined as an array that has more than one dimension. Having more than one dimension means that it can grow in multiple directions.

Q10].Explain string handling functions like strlen(), strcpy(), strcat(),

strcmp(), and strchr(). Provide examples of when these functions are

useful.

A String in C programming is a sequence of characters terminated with a null character ‘\0’. The C String is stored as an array of characters. The difference between a character array and a C string is that the string in C is terminated with a unique character ‘\0’.

Function Description

Syntax

strlen() Find the length of a string excluding ‘\0’ NULL character. strlen(str);

strcpy() Copies a string from the source to the destination. strcpy(dest, src);

strncpy() Copies n characters from source to the destination. strncpy( dest, src, n );

strcat() Concatenate one string to the end of another. strcat(dest, src);

strncat() Concatenate n characters from the string pointed to by src to the end of the string pointed to by dest. strncat(dest, src, n);

strcmp() Compares these two strings lexicographically. strcmp(s1, s2);

strncmp() Compares first n characters from the two strings lexicographically.

strncmp(s1, s2, n);

strchr() Find the first occurrence of a character in a string. strchr(s, c);

strrchr() Find the last occurrence of a character in a string. strchr(s, ch);