DSA SQUAD BY SOURCIFY IN

DAY 1:

INPUT/OUTPUT:

*Inputs and outputs are the fundamentals of any programming language. The word input means the **signals/data** received by the system and the word output means the **signals /data** sent by the system to the user. Generally, input and output syntaxes are different for different programming languages.*

*In this article, we are going to discuss the input/output of c programming language. It as standard libraries that allow input and output in a program. The **stdio.h** or standard input output library in C that has methods for input and output.*

Scanf ()

*This function in c reads the input from the console as per the format specified and stores the value in the given address of the memory. Now, when we say "as per the format specified", we mean there are **few format specifiers** which are in-built in c programming. The format specifier in C is used to tell the compiler about the type of data to be printed or scanned in input and output operations. They always start with a % symbol and are used in the formatted string in functions like printf(), scanf(), sprintf(), etc.*

Some of the commonly used format specifiers are mentioned below:

%c	*For character type.*
%d	*For signed integer type.*
%f	*For float type.*
%ld or %li	*Long*
%Lf	*Long double*
%p	*Pointer*
%s	*String*
%u	*Unsigned int*

Now, lets us jump into the syntax of scanf ()

Syntax:

scanf("%X", &variableOfXType);

where %X is the format specifier in C. The '&' is the address operator and stores the address of the input in the memory. The 'variableOfXType' accepts the input from the console.

Example: For integer

scanf(“%d”, &X);

Where, X is an integer type variable.

For string ,

scanf(“%s”, &Str);

Where, Str is a string type variable.

For character,

scanf(“%c”, &ch);

Where, ch is a character type variable.

Printf()

The printf() statement prints the output on the screen.

The syntax for the printf() statement is as follows:

Syntax:

printf("%X", variableOfXType);

where %X is the format specifier in C and ‘variableOfXType’ is the value to be printed.

Loops and conditional statements:

Loops: *Loops are generally used to execute a particular task a repeated no of times. In C programming there are mainly three types of loops.*

For loop: *The **for loop** in C Language provides a functionality/feature to repeat a set of statements a defined number of times. The for loop is in itself a form of an **entry-controlled loop**. The for loop contains the initialization, condition, and updating statements as part of its syntax. It is mainly used to traverse arrays, vectors, and other data structures.*

Syntax of for Loop

*for(**initialization; check/test expression; increment/decrement**)*

{

// body

}

The working of for loop is mentioned below:

- ***Step 1:** Initialization is the basic step of for loop. This step occurs only once during the start of the loop. During Initialization, variables are declared, or already existing variables are assigned some value.*
- ***Step 2:** During the Second Step condition statements are checked and only if the condition is the satisfied loop we can further process otherwise loop is broken.*
- ***Step 3:** All the statements inside the loop are executed.*
- ***Step 4:** After that, update the variable as defined in the for loop.*
- ***Step 5:** Continue to Step 2 till the loop breaks.*

***While loop:** The **while Loop** is an entry-controlled loop in C programming language. This loop can be used to iterate a part of code while the given condition remains true.*

Syntax for while loop

```
while (test expression)
{
  // body
}
```

We can understand the working of the while loop by following steps :

STEP 1: When the program first comes to the loop, the test condition will be first evaluated.

1. *STEP 2A: If the test condition is false, the body of the loop will be skipped program will continue.*

2. *STEP 2B: If the expression evaluates to true, the body of the loop will be executed.*

3. *STEP 3: After executing the body, the program control will go to STEP 1. This process will continue till the test expression is true.*

Do..while loop: *The **do..while in C** is a loop statement used to repeat some part of the code till the given condition is fulfilled. It is a form of an **exit-controlled or post-tested loop** where the test condition is checked after executing the body of the loop. The do..while is executed atleast once even if the condition is false.*

Syntax of do…while Loop

do *{*

// body

} ***while*** *(condition);*

The working of the do…while loop is explained below:

1. *When the program control first comes to the do…while loop,* ***the body of the loop is executed first and then the test condition/expression is checked****, unlike other loops where the test condition is checked first. Due to this*

property, the do…while loop is also called exit controlled or post-tested loop.

2. *When the test condition is evaluated as **true**, the **program control goes to the start** of the loop and the body is executed once more.*
3. *The above process repeats till the test condition is true.*
4. *When the test condition is evaluated as **false, the program controls move on to the next statements** after the do…while loop.*

Jump statements: *Jump statements are used to jump from one part of the code to another altering the normal flow of the program. Generally, in c, there are four types of jump statements:*

1. *break*
2. *continue*
3. *goto*
4. *return*

Break: *The break statement exits or terminates the loop or switch statement based on a certain condition, without executing the remaining code. It is only applicable to loops or switch statements. Break statement does not work on conditional statements.*

Syntax:

```
break;
```

Continue: *The **C continue statement** resets program control to the **beginning** of the loop when encountered. As a result, the current iteration of the loop gets skipped and the control moves on to the next iteration. Statements after the continue statement in the loop are not executed.*

Syntax:

```
Continue;
```

Goto: *The goto statement can be used to jump from anywhere to anywhere within a function*.

Return: *The return statement is generally used to return a something to the function if it is not of void type*.