

1.) Discuss the concept of if - else - if , if - else - ladder and nested if .

Ans → An if - else - if statement is a type of control flow statement that allows a program to make decisions based on multiple conditions . It consists of one or more if statements , each followed by an optional else if statement and an optional else statement . The if statements are evaluated in order and the first one whose condition is satisfied is executed . If none of the conditions are met the else statement , if present , is executed .

The if - else - ladder is a type of control flow statement that is used programming to determine which of a set of conditions is true and executed a block of code based on the result . It consists a series of if - else statements , which are evaluated in order from top to bottom . If the condition of the top of ladder is true , then the associated code is executed . And the rest of the ladder is ignored .

It is always legal in C to nest if - else statements , which means you can use one if or else if statement inside another if or else if statement (s) and you can nest else if ... else in the similar way as you have nested if statement .

2.) Discuss the concept of switch case ?

Ans → A switch statement allows a variable to be tested for equality against a list of values . Each value is called a case , and the variable being switched on is checked for each switch case .

Syntax: → switch (expression)

```
{  
    case constant expression:  
        statement(s);  
        break; // optional;  
    case constant expression:  
        statement(s);  
        break; // optional  
    default: /* optional */  
        statement(s);  
}
```

Rules to apply switch statement →

- The expression used in a switch statement must have an integral or enumerated type.
- You can have any number of case statement within a switch. each case followed by the value to be compared to a colon.
- The constant expression of a case must be the same data type as the variable in switch , and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached. etc
- A switch statement can have an optional default case, which must appear at the end of the switch.

3. b Explain the concept of GOTO in C with suitable example.

Ans → A goto statement in C language provides an unconditional jump from the goto to a labeled statement in the same function.

The given label must reside in the same function.

Syntax → goto label;

..

label: statement;

Here label can any plain text except C keyword and it can be set anywhere in the C program above or below to goto statement.

Code example → int main()

```
{ /* Local variable definition */
    int a = 10;
    LOOP: do
    {
        if (a == 15)
        {
            a = a + 1;
            goto LOOP;
        }
        printf("value of a: %d\n", a);
        a++;
    } while (a < 0);
    return 0;
}
```

Output:

Value of a : 10
Value of a : 11
Value of a : 12
Value of a : 13
⋮
Value of a : 19

4.) What is difference between while loop and do while loop?

Parameters	while	do-while
Checking of condition	It first needs to check the condition and only then can we execute the statement(s).	The execution of statement(s) occurs at least once. After that, the checking of condition occurs.
Semicolon	while (condition) No semicolon is present at the end of while loop.	while (condition); A semicolon is present at the end.
Requirements of brackets	We don't require any brackets if there is only a single statement.	We could always require brackets in this case.
Controlling	The loop is entry controlled type of loop.	The do-while loop is an exit-controlled type of loop.

5.) What will be the output:

```
int main ()  
{  
    int i, j;  
    for (i=1, j=i+1; i<20, j<20; i+=2, j+=2)  
    {  
        printf ("%d %d \n", i, j);  
    }  
}
```

Output:-

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18

A2 →

6.) Find the errors of the following code:
Rewrite the above code in correct form. What will be the output of the corrected code if $a=5$, $b=10$ is assigned as the input?

Aus → Code after fixing all errors :→

```
#include <stdio.h>
int main() {
    int a, b, temp;
    printf("Enter the value of a and b: ");
    scanf("%d %d", &a, &b);
    temp = a;
    a = b;
    b = temp;
    printf("%d %d", a, b);
    return 0;
}
```

Output of the code after $a=5$, $b=10$ is assigned as the input →

Output:- 10 5

7.) Write a C program to calculate electricity bill using if - else statement.

For first 50 units — Rs. 1.00/unit

Next 100 units — Rs. 5.50/unit

Next 100 units — Rs. 6.00/unit

Above 250 — Rs. 6.50/unit

Aus →

```
#include <stdio.h>
int main()
{
    int unit;
    float total;
    printf("Enter total units consumed : ");
    scanf("%d", &unit);
    if (unit <= 50)
    {
        total = unit * 4.00;
        printf("your total bill is = %.2f ", total);
    }
    else if (unit <= 150)
    {
        total = (50 * 4.00) + ((unit - 50) * 5.50);
        printf("your total bill is = %.2f ", total);
    }
    else if (unit <= 250)
    {
        total = (50 * 4.00) + (100 * 5.50) + ((unit - 150) * 6.00);
        printf("your total bill is = %.2f ", total);
    }
    else
    {
        total = (50 * 4.00) + (100 * 5.50) + (100 * 6.00) + ((unit - 250)
            * 6.50);
        printf("your total bill is = %.2f ", total);
    }
    return 0;
}
```

8.) Write a program in C print the multiplication table for an input (input may be any number from 1 to 10).

Ans →

```
#include <stdio.h>
int main()
{
    int num, i;
    printf("Enter no. between 1 to 10: ");
    scanf("%d", &num);
    for (i=1; i<=10; i++)
    {
        printf("%d x %d = %d\n", num, i, num*i);
    }
    return 0;
}
```

9.) Discuss the concept of nested for loop with example.

Ans → C for loop contains the initialization and update of loop control variables in the syntax itself. Nested loop means a loop inside another loop. It works with all three loops but mostly it can be used with for loop. Inner loop will execute some of the statements and the outer loop will execute the inner loop depends on given conditions.

```
#include <stdio.h>
int main()
{
    int i, j;
    for (i=0; i<=4; i++)
    {
        for (j=0; j<=i; j++)
        {
```

```

        printf("* ");
    }
    printf("\n");
}
return 0;
}

```

Output :-

*				
*	*			
*	*	*		
*	*	*	*	
*	*	*	*	*

10) What is array? Discuss different types of arrays available in C.

Ans → An array is a continuous memory location which stores similar type of data. It has ability to use a single name to represent a collection of data items that enables us to develop concise and efficient program. It is a linear data structure, which is used represent various data structure like STACK, QUEUE, and TREES etc.

Some examples → 1. List of employees in an organization
2. Table of daily rainfall data etc.

We can use array to represent not only simple lists of values but also tables of data in two, three or more dimensions. So, there are three types of arrays.

These are →

1. One-dimensional Array
2. Two-dimensional Array
3. Multi-dimensional Array