Module: 2

Day3:
Trainer: Asif Nafees

For Loop:
In C, a for loop is a control structure that allows you to repeat a block of code a certain number of times. It is commonly used when the number of iterations is known beforehand. The syntax of the for loop provides a concise way to initialize a loop variable, test a condition, and update the loop variable, all in a single line.

Syntax of for Loop

```
for (initialization; condition; increment/decrement) {
    // Body of the loop
}
```

initialization: This step allows you to initialize a loop control variable (like int i = 0;). This part is executed only once at the start of the loop.
condition: The condition is evaluated before every iteration of the loop. If the condition is true, the loop body is executed. If it is false, the loop terminates.
increment/decrement: This step modifies the loop control variable after each iteration. It could be an increment (i++) or a decrement (i--), depending on the requirement.

Flow of Control
1. The initialization is executed. Typically, this is used to set a counter variable.
2. The condition is checked. If it's true, the loop body is executed. If false, the loop terminates.
3. After executing the loop body, the increment/decrement step is executed.
4. The condition is checked again, and the process repeats.

Example of a Basic for Loop

```
#include <stdio.h>
int main() {
    int i;
    // Loop from 0 to 4
    for (i = 0; i < 5; i++) {
        printf("Iteration %d\n", i);
    }
    return 0;
}
```

Output:

```
Iteration 0
Iteration 1
Iteration 2
```

Iteration 3
Iteration 4

Explanation:
The initialization int i = 0 sets the variable i to 0.
The condition i < 5 checks if i is less than 5 before each iteration.
After printing the value of i, the increment i++ increases the value of i by 1.
The loop runs as long as the condition is true.

Variations of for Loop
1. Multiple Initialization or Increment
You can initialize or increment multiple variables in a for loop by separating them with commas.

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 0, j = 5; i < j; i++, j--) {
        printf("i = %d, j = %d\n", i, j);
    }
    return 0;
}
```

Output:
i = 0, j = 5
i = 1, j = 4
i = 2, j = 3

2. Omitting Initialization or Increment
You can omit the initialization or increment if needed.

```
#include <stdio.h>
int main() {
    int i = 0;
    for (; i < 5; i++) {
        printf("i = %d\n", i);
    }
    return 0;
}
```

Infinite for Loop
A for loop can become an infinite loop if the condition never becomes false.

```
for (;;) {
    // Infinite loop
}
```

This loop runs forever because there's no condition to stop it.
Use of break and continue in for Loop
1. break Statement
The break statement is used to exit the loop prematurely, regardless of the loop's condition.

```
#include <stdio.h>
int main() {
    int i;
    for (i = 0; i < 10; i++) {
        if (i == 5) {
            break; // Loop will terminate when i equals 5
        }
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

```
0
1
2
3
4
```

2. continue Statement
The continue statement skips the rest of the current iteration and moves on to the next iteration.

```
#include <stdio.h>
int main() {
    int i;
    for (i = 0; i < 10; i++) {
        if (i % 2 == 0) {
            continue; // Skip even numbers
        }
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

```
1
3
5
7
9
```

Assignment

1. Simple Count Loop (1 to 10):

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```

2. Loop in Reverse (10 to 1):

```
#include <stdio.h>
int main() {
    for(int i = 10; i >= 1; i--) {
        printf("%d\n", i);
    }
    return 0;
}
```

3. Even Numbers (1 to 20):

```
#include <stdio.h>
int main() {
    for(int i = 2; i <= 20; i += 2) {
        printf("%d\n", i);
    }
    return 0;
}
```

4. Odd Numbers (1 to 19):

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 19; i += 2) {
        printf("%d\n", i);
    }
    return 0;
}
```

5. Sum of First 10 Natural Numbers:

```
#include <stdio.h>
int main() {
    int sum = 0;
    for(int i = 1; i <= 10; i++) {
        sum += i;
    }
    printf("Sum = %d\n", sum);
```

```
    return 0;
}
```

6. Factorial of a Number:

```
#include <stdio.h>
int main() {
    int num = 5, factorial = 1;
    for(int i = 1; i <= num; i++) {
        factorial *= i;
    }
    printf("Factorial = %d\n", factorial);
    return 0;
}
```

7. Multiplication Table for 5:

```
#include <stdio.h>
int main() {
    int num = 5;
    for(int i = 1; i <= 10; i++) {
        printf("%d x %d = %d\n", num, i, num * i);
    }
    return 0;
}
```

8.Loop Through Characters (A to Z):

```
#include <stdio.h>
int main() {
    for(char ch = 'A'; ch <= 'Z'; ch++) {
        printf("%c ", ch);
    }
    return 0;
}
```

9. Printing Squares of Numbers (1 to 10):

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 10; i++) {
        printf("Square of %d = %d\n", i, i * i);
    }
    return 0;
}
```