SESSION 4: LOOPS

Loops in Programming :

A **loop** is a control structure that allows a block of code to be executed **repeatedly** as long as a specified condition is true (or until a condition is met).
Loops are mainly used when a task needs to be performed multiple times without writing the same code again and again, making programs more efficient and concise.

Key Points:

- **Initialization** → Setting the starting point.
- **Condition** → Checked before (or after) each iteration.
- **Iteration/Update** → Changes the variable so the loop can progress.
- **Body** → The statements that run repeatedly.

Types of Loops in C

---

1. FOR loop

The for loop is used when the number of iterations is **known in advance**. It executes a block of code repeatedly until a condition is false.

Syntax:

```
for(initialization; condition; update) {
   // code to be executed
}
```

Example:

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

Prints numbers from 1 to 5.

2. WHILE loop

The **while loop** is used when the number of iterations is **not fixed in advance**. It checks the condition **before** executing the loop body.

Syntax:

```
while(condition) {
    // code to be executed
}
```

Example:

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

Prints numbers from 1 to 5.

3. DO-WHILE loop

The **do-while loop** is similar to `while`, but the condition is checked **after** executing the loop body. This ensures the loop body executes at least **once**, even if the condition is false.

Syntax:

```
do {
    // code to be executed
} while(condition);
```

Example:

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

Prints numbers from 1 to 5.

Problems on Loops:

1. Print 1 to N Using Loops

```
#include <stdio.h>
int main() {
   int N;
   printf("Enter N: ");
   scanf("%d", &N);
   for(int i = 1; i <= N; i++) {
      printf("%d ", i);
   }
   return 0;
}
```

2. Palindrome Number

```
bool isPalindrome(int x) {
   int rem = 0;
   long reverse = 0;
   long long temp = x;
   while(x > 0){
    rem = x%10;
    reverse =  reverse*10 + rem;
    x/=10;// x = x/10;
   }
   if(reverse == temp) return true;
   return false;
}
```