Understanding Loops in C Programming

**Objective:**

The objective of this lab is to understand and implement the three main types of loops in C programming: for loop, while loop, and do-while loop. These loops are used to execute a block of code multiple times based on specific conditions.

**Introduction:**

In C programming, loops are control structures used to repeat a block of code multiple times. The primary types of loops in C are:

1. **For Loop**: Used when the number of iterations is known beforehand.
2. **While Loop**: Used when the number of iterations is not predetermined but depends on a condition.
3. **Do-While Loop**: Similar to the while loop but the condition is checked after the execution of the loop body, ensuring the body is executed at least once.

**Types of Loops:**

1. **For Loop**:
   * Syntax:

for(initialization; condition; increment/decrement) {

//statement

}

* + The for loop is best used when the number of iterations is known in advance. It initializes the loop control variable, checks the condition before each iteration, and then increments or decrements the loop variable.

1. **While Loop**:

* Syntax:

while(condition) {

// loop body

}

* A while loop is used when you do not know how many times the loop will execute but you know the condition that must be true for the loop to run

1. **Do-While Loop**:

* Syntax:

do {

// loop body

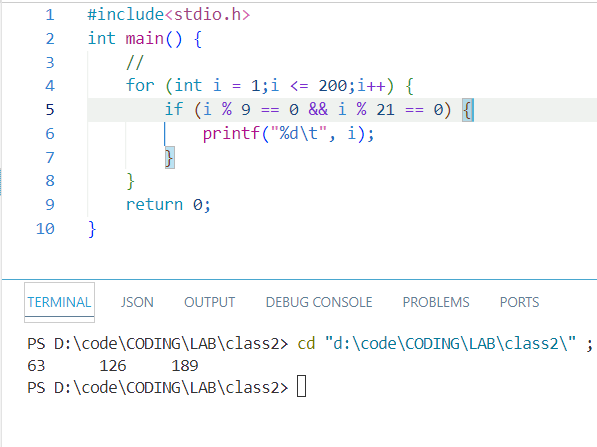
} while(condition);

* The do-while loop is similar to the while loop, but the condition is evaluated after the loop body executes. This guarantees that the loop will execute at least once, even if the condition is initially false.

1. **For loop**
   1. **Write a C program to print every second alphabet starting from 'A' to 'Z'.**

****

* 1. **Write a C program to print all numbers between 1 and 200 that are divisible by both 9 and 21.**

****

* 1. **Write a C program to calculate the factorial of a given number.**

****

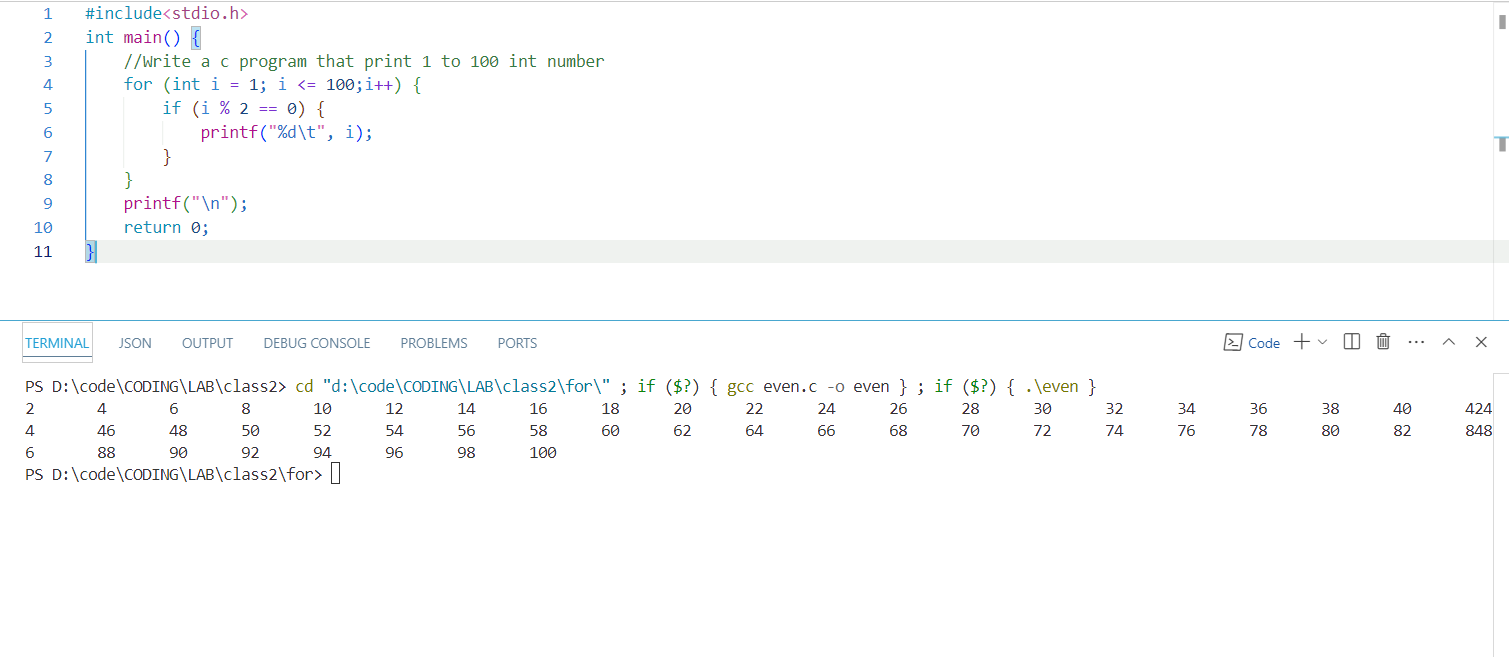
* 1. **Write a C program to calculate the sum of squares of numbers from 1 to n.**

****

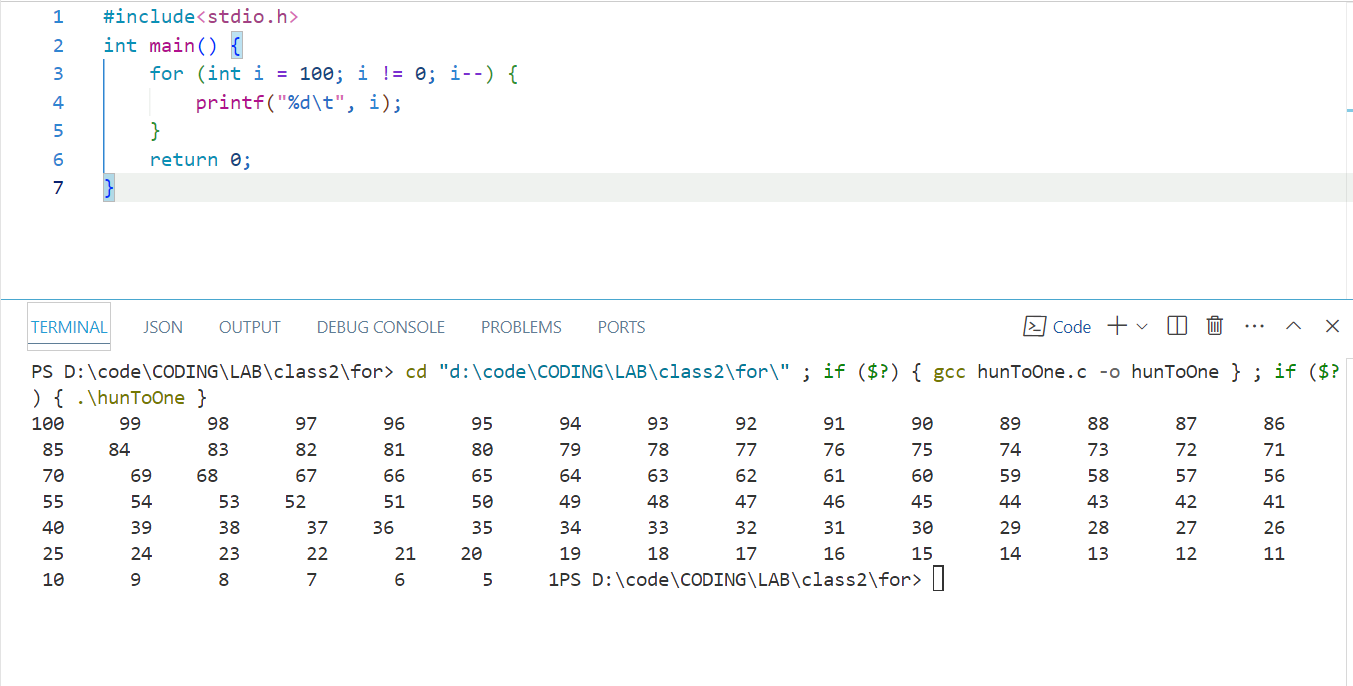
* 1. **Write a C program to calculate the sum of numbers from 1 to n.**

****

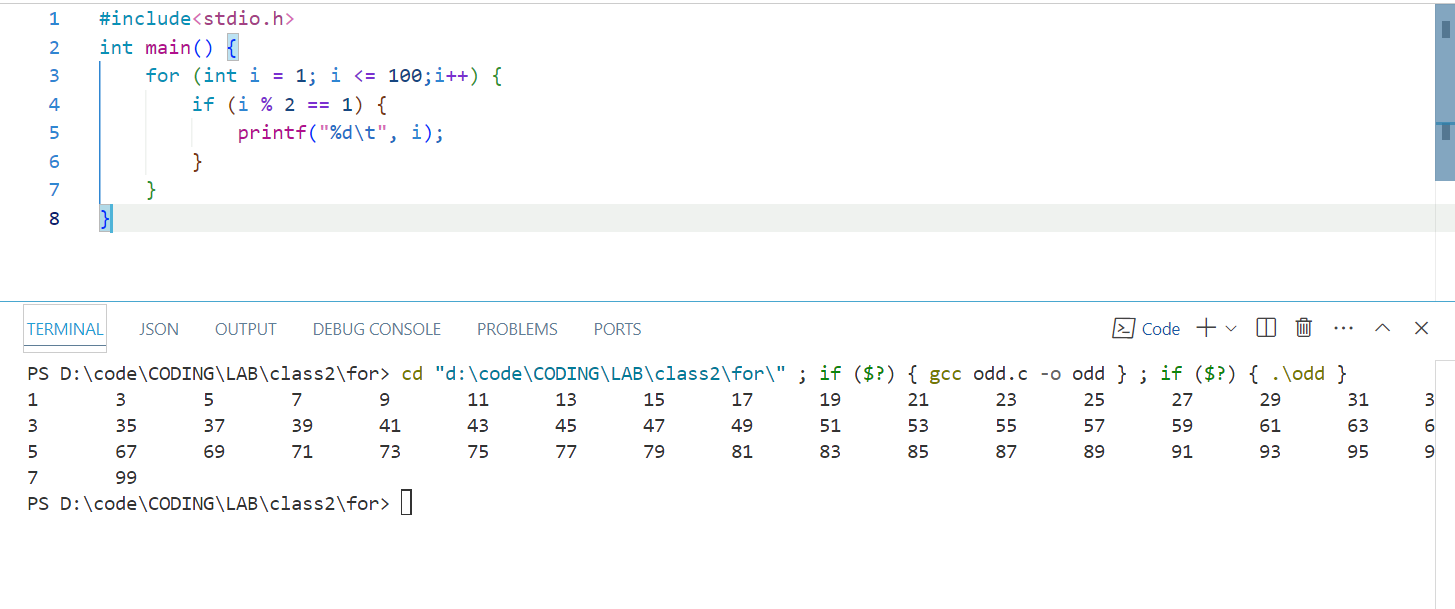
* 1. **Write a C program to print all even numbers from 1 to 100.**

****

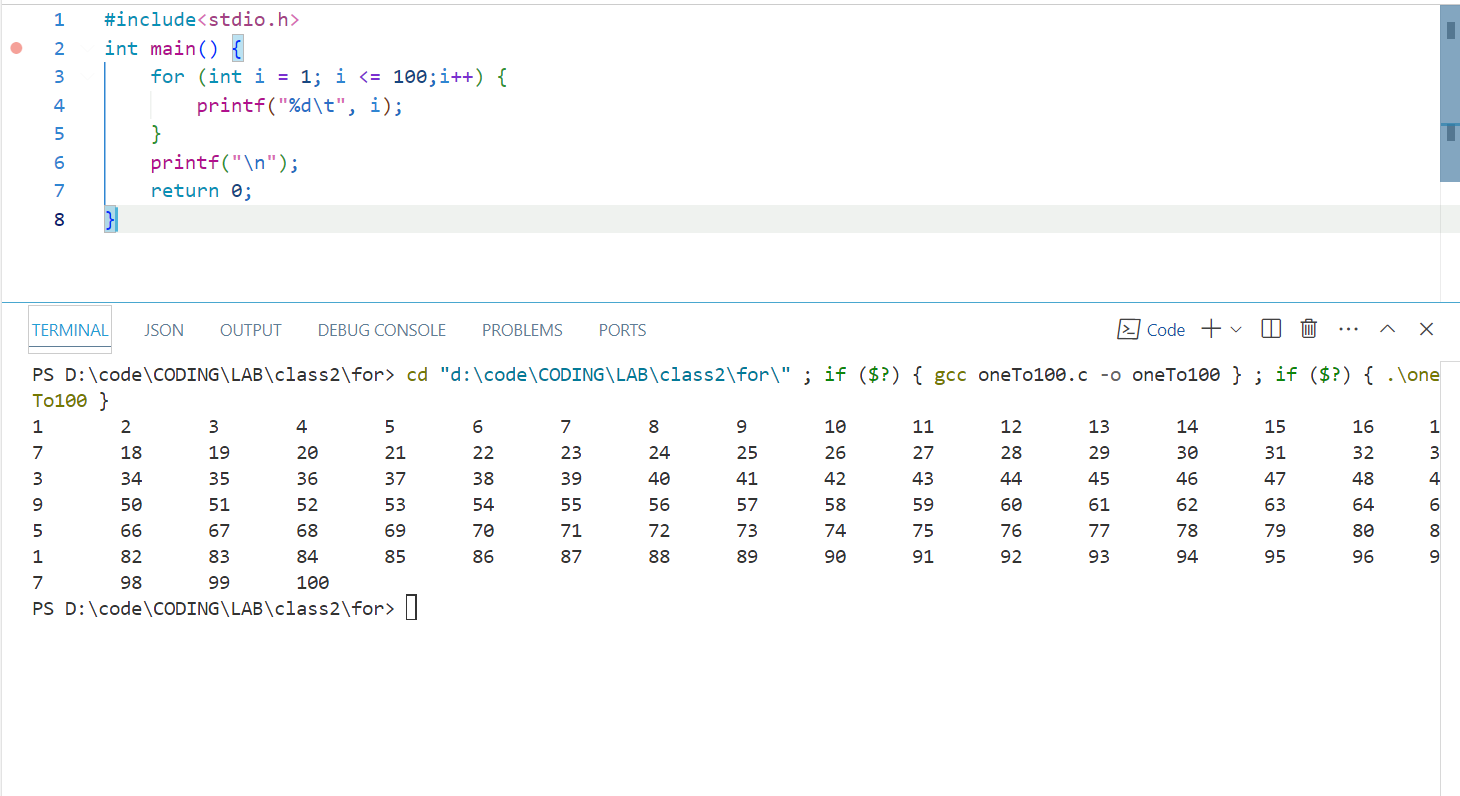
* 1. **Write a C program to print numbers from 100 to 1 in descending order.**

****

* 1. **"Write a C program to print all odd numbers from 1 to 100."**

****

* 1. **Write a C program to print numbers from 1 to 100.**

****

**Conclusion:**

In this lab, we explored the three main loop constructs in C: the for, while, and do-while loops. Each loop has its specific use case based on how and when the condition is evaluated. The understanding of loops is fundamental in solving repetitive tasks efficiently in C programming. By using loops, we can minimize code duplication and automate repetitive tasks.