

# Nested Loops in C Programming

Biswajit Borgohain

Roll No: 14, MCA 1st

November 12, 2024

# Contents

- 1 Introduction to Nested Loops
- 2 Memory Representation
- 3 Real-Life Example
- 4 Output of the Example
- 5 Implementation in C
- 6 Output of Pattern Code
- 7 Advantages of Nested Loops
- 8 Limitations of Nested Loops
- 9 Conclusion

# What are Nested Loops?

- **Nested loops** are loops inside another loop. The inner loop is executed completely every time the outer loop runs one iteration.
- Used for handling multi-dimensional data, repetitive patterns, and complex data processing.
- Example Syntax:

## Nested Loop Syntax

```
for (initialization; condition; update) {  
    for (initialization; condition; update) {  
        // Code to execute  
    }  
}
```

# How Nested Loops Work in Memory

- **Execution Flow:** The outer loop begins execution, and for each iteration of the outer loop, the inner loop runs to completion.
- **Memory Pattern:** Imagine each iteration of the outer loop as a level; within each level, the inner loop iterates completely before the next outer loop level begins.

## Visualization of Nested Loop Execution

- **Example:** Suppose we have an outer loop running 3 times and an inner loop running 2 times:
  - Outer Loop Iteration 1:
    - Inner Loop Iteration 1
    - Inner Loop Iteration 2
  - Outer Loop Iteration 2:
    - Inner Loop Iteration 1
    - Inner Loop Iteration 2
  - Outer Loop Iteration 3:
    - Inner Loop Iteration 1
    - Inner Loop Iteration 2

# Real-Life Example of Nested Loops

- **Example: Generating Multiplication Tables**
- Using nested loops to print multiplication tables from 1 to 5.

## C Code Example

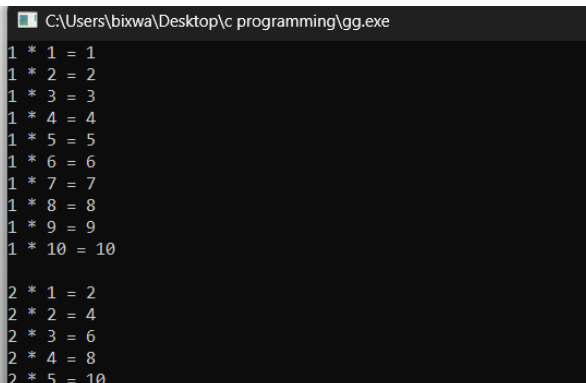
```
#include <stdio.h>

int main() {

    int i, j;
    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= 10; j++) {
            printf("%d * %d = %d\n", i, j, i * j);
        }
        printf("\n"); // New line for each table
    }
    return 0;
}
```

# Output of Multiplication Table Code

- The output of the code will display multiplication tables from 1 to 5.
- Space reserved for image showing the output in the console.

A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\bixwa\Desktop\c programming\gg.exe". The console output displays multiplication tables for numbers 1 through 5. Each table consists of five lines, one for each multiplier from 1 to 5. The first table (for 1) shows 1 \* 1 = 1 up to 1 \* 10 = 10. The second table (for 2) shows 2 \* 1 = 2 up to 2 \* 5 = 10. The third table (for 3) shows 3 \* 1 = 3 up to 3 \* 5 = 15. The fourth table (for 4) shows 4 \* 1 = 4 up to 4 \* 5 = 20. The fifth table (for 5) shows 5 \* 1 = 5 up to 5 \* 5 = 25. The output is as follows:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
```

# Implementing Nested Loops in C

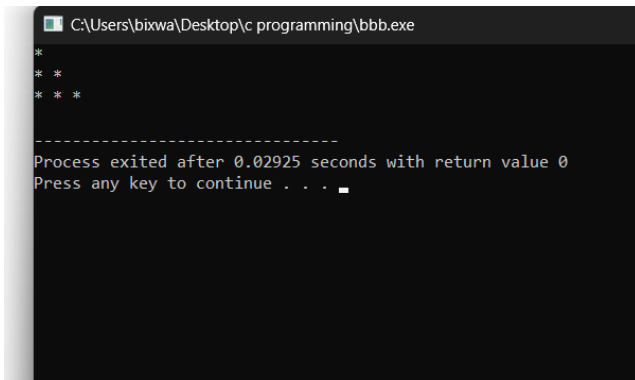
- Example code for creating a pattern using nested loops.

## Example Code

```
#include <stdio.h>
int main() {
    int n = 3, i, j;
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

# Output of Pattern Code

- The output of the code will display a pattern as shown.
- Space reserved for image showing the output in the console.



```
C:\Users\bixwa\Desktop\c programming\bbb.exe
*
* *
* * *

-----
Process exited after 0.02925 seconds with return value 0
Press any key to continue . . .
```



# Advantages of Using Nested Loops

- Efficiently handles multi-dimensional data (e.g., matrices).
- Useful for generating patterns or tables.
- Allows multiple layers of operations on data sets.

# Limitations of Nested Loops

- Can be computationally expensive for large datasets.
- Higher nesting levels can make code complex and difficult to read.
- May lead to inefficient memory usage in certain cases.

# Conclusion

- Nested loops are essential for processing complex data structures and patterns.
- Widely used in real-world applications like table generation, pattern creation, and matrix operations.
- Use nested loops judiciously to balance readability and efficiency.

# Thank You!