AL-KAWTHAR UNIVERSITY

Department of Computer Science

CS 121 L – Programming Fundamentals (PF)

Lab # 05

Objective:

To introduce students to the concept of nested loops in C programming. Students will learn how to use nested loops to solve problems involving patterns, palindromes, and other iterative tasks.

Name of Student	**Muhammad Hashir Rafique**
Student ID	**BSCS24101023**
Date of Lab Conducted	
Marks Obtained	
Remarks	
Signature	

LAB 5 - ACTIVITY 1

Drawing 2D Patterns using Nested Loops

Objective:

- Demonstrate the use of nested loops to print 2D patterns.

Activities:

1. Write a C program to print the following pattern using nested loops:

```
*
* *
* * *
* * * *
* * * * *
```

Example Code:

```
#include <stdio.h>

int main() {
    int rows = 5;

    for (int i = 1; i <= rows; i++) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

2. Modify the program to print the following inverted pattern:

```
* * * * *
* * * *
* * *
* *
*
```

3. Dry-run the programs with sample values and explain the logic.

- -

1. Pattern Printing

```
#include <stdio.h>

int main() {
    int rows = 5;

    for (int i = 1; i <= rows; i++) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

Dry Run:

```
*
* *
* * *
* * * *
* * * * *
```

Logic - In this pattern, we use two nested loops.

- The outer loop runs from 1 to the number of rows (which is 5), and it controls the number of lines printed.
- The inner loop is responsible for printing the asterisks.
- For each row `i`, the inner loop runs from 1 to `i`, printing one `*` followed by a space each time. So, in the first row, it prints one star, in the second row two stars, and so on.
- After each row is printed, a newline character `\n` moves the cursor to the next line. This creates the right-angled triangle growing downward.

2. Inverted Pattern Printing

```
#include <stdio.h>

int main() {
    int rows = 5;

    for (int i = rows; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n"); }

    return 0;
```

```
}
```

Dry Run:

```
* * * * *
* * * *
* * *
* *
*
```

Logic - In this inverted pattern, the structure is similar, but the outer loop counts down instead of up.

- It starts from 5 and decreases to 1, which controls how many lines will be printed.
- The inner loop again prints the asterisks, but this time it prints fewer stars with each row.
- In the first iteration, it prints five stars because `i` is 5. In the next iteration, `i` becomes 4, so it prints four stars, and this continues until it reaches 1 star.
- After each set of stars, a newline is printed to move to the next line.
- This creates an upside-down right-angled triangle.

LAB 5 - ACTIVITY 2

Palindrome Checker using Nested Loops

Objective:

- Use nested loops to check if a given string is a palindrome.

Activities:

1. Define a palindrome (a word that reads the same backward as forward, e.g., "madam").

Example Code:

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[100];

    printf("Enter a string: ");
    scanf("%s", str);

    int len = strlen(str);
    int is_palindrome = 1;

    for (int i = 0, j = len - 1; i < j; i++, j--) {
        if (str[i] != str[j]) {
            is_palindrome = 0;
            break;
        }
    }

    if (is_palindrome) {
        printf("%s is a palindrome.\n", str);
    } else {
        printf("%s is NOT a palindrome.\n", str);}

    return 0;}
```

2. Write a C program to check if a user-input string is a palindrome using nested loops.
3. Test the program with sample inputs (e.g., "racecar", "hello").

C Language Code

```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[100];
    int is_palindrome = 1;

    printf("Enter a string: ");
    scanf("%s", str);

    int len = strlen(str);

    // comparing front and back characters
    for (int i = 0, j = len - 1; i < j; i++, j--) {
        if (str[i] != str[j]) {
            is_palindrome = 0;
            break;
        }
    }

    if (is_palindrome) {
        printf("%s is a palindrome.\n", str);
    } else {
        printf("%s is NOT a palindrome.\n", str);
    }

    return 0;
}
```

Dry Run:

```
Enter a string: civic
civic is a palindrome.
```

LAB 5 - ACTIVITY 3

Multiplication Table using Nested Loops

Objective:

- Generate a multiplication table using nested loops.

Activities:

1. Write a C program to print the multiplication table from 1 to 10 using nested loops.
2. Format the output neatly, like below:

```
1 x 1 = 1
1 x 2 = 2
...
10 x 10 = 100
```

3. Dry-run the program and explain the loop iterations.

Example Code:

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
        }
        printf("\n");
    }

    return 0;
}
```

- -

C Language program

```
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            printf("%d x %d = %d\n", i, j, i * j);
        }
        printf("\n"); // Adds a blank line after each table
    }

    return 0;
}
```

Dry Run:

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8

…
…
…

10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
```

LAB 5 - ASSIGNMENT

Prepare a document that includes the following:

1. Definition: Nested loops involve loops inside another loop, useful for grids/multi-level iterations.
2. Programs in C language:
    - Pattern: Print a pyramid using *.
    - Palindrome: Check if a number (e.g., 121) is a palindrome.
    - Grid: Print a 5x5 grid of numbers (1 to 25).
3. Flowcharts: Hand-drawn for each program.

---

(1)

Definition: Nested Loops in C

A nested loop means using one loop inside another loop. It's commonly used when you need to work with multi-dimensional patterns, such as printing grids, patterns, or handling rows and columns. For example, to print a 5x5 grid, you use an outer loop for rows and an inner loop for columns.

(2)

Pyramid Code:

```
#include <stdio.h>

int main() {
    int rows = 5;

    for (int i = 1; i <= rows; i++) {
        // Print spaces
        for (int space = 1; space <= rows - i; space++) {
            printf(" ");
        }
        // Print stars
        for (int j = 1; j <= 2 * i - 1; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

FLOWCHART OF THE PREVIOUS PROGRAM

Palindrome checker:

```
#include <stdio.h>

int main() {
    int num, original, reversed = 0, digit;

    printf("Enter a number: ");
    scanf("%d", &num);

    original = num;

    while (num != 0) {
        digit = num % 10;
        reversed = reversed * 10 + digit;
        num = num / 10;
    }

    if (original == reversed) {
        printf("%d is a palindrome.\n", original);
    } else {
        printf("%d is NOT a palindrome.\n", original);
    }

    return 0;
}
```

Dry Run:

```
Enter a number: 678
678 is NOT a palindrome.
```

FLOWCHART OF THE PREVIOUS PROGRAM

Grid:

```
#include <stdio.h>

int main() {
    int count = 1;

    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            printf("%2d ", count);
            count++;
        }
        printf("\n");
    }

    return 0;
}
```

Dry Run:

```
 1  2  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

FLOWCHART OF THE PREVIOUS PROGRAM