

Module: 2

Day4:

Trainer: Asif Nafees

In C programming, nested loops occur when one loop is placed inside another. You can nest different types of loops, such as while inside for, for inside while, and so on. Here's an explanation of how nested while and for loops work with examples.

1. Nested while Loops:

A while loop can be placed inside another while loop. The inner loop completes all its iterations before the outer loop moves to the next iteration.

Syntax:

```
while(condition1) {  
    // Outer loop body  
    while(condition2) {  
        // Inner loop body  
    }  
}
```

Example:

```
#include <stdio.h>  
int main() {  
    int i = 1;  
    // Outer while loop  
    while(i <= 3) {  
        int j = 1;  
        // Inner while loop  
        while(j <= 3) {  
            printf("i = %d, j = %d\n", i, j);  
            j++;  
        }  
        i++;  
    }  
    return 0;  
}
```

Output:

```
i = 1, j = 1  
i = 1, j = 2  
i = 1, j = 3  
i = 2, j = 1  
i = 2, j = 2  
i = 2, j = 3
```

```
i = 3, j = 1  
i = 3, j = 2  
i = 3, j = 3
```

In this example:

The outer loop runs 3 times with i taking values from 1 to 3.

For each value of i, the inner loop runs 3 times, with j also taking values from 1 to 3.

2. Nested for Loops:

Similarly, you can nest a for loop inside another for loop. The inner loop will execute completely for each iteration of the outer loop.

Syntax:

```
for(initialization; condition1; update) {  
    // Outer loop body  
    for(initialization; condition2; update) {  
        // Inner loop body  
    }  
}
```

Example:

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

Output:

```
i = 1, j = 1  
i = 1, j = 2  
i = 1, j = 3  
i = 2, j = 1  
i = 2, j = 2  
i = 2, j = 3  
i = 3, j = 1  
i = 3, j = 2  
i = 3, j = 3
```

In this case:

The outer loop runs 3 times (i from 1 to 3).

The inner loop runs 3 times for each outer loop iteration (j from 1 to 3).

3. Nested for and while Loops:

You can also nest a for loop inside a while loop (or vice versa). The principle remains the same: the inner loop completes all its iterations before the outer loop proceeds to the next iteration.

Example of while inside for:

```
#include <stdio.h>
int main() {
    // Outer for loop
    for(int i = 1; i <= 3; i++) {
        int j = 1;
        // Inner while loop
        while(j <= 3) {
            printf("i = %d, j = %d\n", i, j);
            j++;
        }
    }
    return 0;
}
```

Example of for inside while:

```
#include <stdio.h>
int main() {
    int i = 1;
    // Outer while loop
    while(i <= 3) {
        // Inner for loop
        for(int j = 1; j <= 3; j++) {
            printf("i = %d, j = %d\n", i, j);
        }
        i++;
    }
    return 0;
}
```

Assignment

1. Square of Stars

```
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the size of the square: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

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

2. Right-Angle Triangle of Stars

```
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

```
*
```

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

3. Inverted Right-Angle Triangle of Stars

```
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = n; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

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

4. Pyramid Pattern

```
#include <stdio.h>
int main() {
    int n, i, j, space;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        for (space = 1; space <= n - i; space++) {
            printf(" ");
        }
        for (j = 1; j <= 2 * i - 1; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

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

5. Inverted Pyramid Pattern

```
#include <stdio.h>  
int main() {  
    int n, i, j, space;  
    printf("Enter the number of rows: ");  
    scanf("%d", &n);  
    for (i = n; i >= 1; i--) {  
        for (space = 1; space <= n - i; space++) {  
            printf(" ");  
        }  
        for (j = 1; j <= 2 * i - 1; j++) {  
            printf("* ");  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

Input:

5

Output:

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

6. Floyd's Triangle

```
#include <stdio.h>  
int main() {  
    int n, i, j, num = 1;  
    printf("Enter the number of rows: ");  
    scanf("%d", &n);
```

```

for (i = 1; i <= n; i++) {
    for (j = 1; j <= i; j++) {
        printf("%d ", num);
        num++;
    }
    printf("\n");
}
return 0;
}

```

Input:

5

Output:

```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```

7. Diamond Pattern

```

#include <stdio.h>
int main() {
    int n, i, j, space;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    // Upper part
    for (i = 1; i <= n; i++) {
        for (space = 1; space <= n - i; space++) {
            printf(" ");
        }
        for (j = 1; j <= 2 * i - 1; j++) {
            printf("* ");
        }
        printf("\n");
    }
    // Lower part
    for (i = n - 1; i >= 1; i--) {
        for (space = 1; space <= n - i; space++) {
            printf(" ");
        }
        for (j = 1; j <= 2 * i - 1; j++) {
            printf("* ");
        }
        printf("\n");
    }
}

```

```
    }
    return 0;
}
```

Input:

5

Output:

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

8. Number Triangle Pattern

```
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

9. Inverted Number Triangle

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

```

int main() {
    int n, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = n; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}

```

Input:

5

Output:

```

1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

```

10. Pascal's Triangle

```

#include <stdio.h>
int main() {
    int n, i, j, space, num;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        for (space = 0; space < n - i; space++) {
            printf(" ");
        }
        num = 1;
        for (j = 0; j <= i; j++) {
            printf("%4d", num);
            num = num * (i - j) / (j + 1);
        }
        printf("\n");
    }
    return 0;
}

```

Input:

5

Output:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

11. Half Pyramid of Numbers

```
#include <stdio.h>
int main() {
    int n, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= i; j++) {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
```

Input:

5

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```