

Dhaka International University



DEPARTMENT OF CSE

LAB REPORT

COURSE NAME : Structured Programming Language Lab

COURSE CODE : 0613-102

REPORT NO : 10

REPORT ON : Advanced Patterns Using loops in C
Programming

<u>SUBMITTED BY</u>	<u>SUBMITTED TO</u>
NAME : Mir Yeasir Abram	Raleib Mahmud
ROLL : 35	Lecturer
REG. NO : CS-D-98-23-127358	Dept. of CSE
BATCH : 08(1st shift)	
SEMESTER : 1st	

DATE OF SUBMISSION : 30 November, 2024

Date of Performance: 23 November, 2024

■ Title: Advanced Patterns Using Loops in C Programming.

■ Objective:

01. To understand and implement various loop structures in c programming to create pattern based output.
02. To explore the applications of nested loops for generating symmetrical, triangular and other structured outputs using numbers, characters and symbols.

■ Introduction:

Pattern generation using loops is a common application in programming . especially for understanding nested loops, logic building , and algorithm design . By combining for loops with conditional statements it becomes possible to create intricate patterns like triangles, diamonds and pyramids.

Each pattern demonstrate a different approach to structuring rows and columns, along with spacing and alignment using nested loops. These programs help in visualizing concepts like iteration and conditional execution , which are fundamental in programming .

■ Discussion:

In this section we will learn about presentation of code and output of code. The shown code will be as in IDE and the output will be as in console.

01. Pattern 2 type - 01 : 1 2 3
1 2
1

... code:

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);

    for (int row = n; row >= 0, row--)
    {
        for (int col = 1; col <= row; col++)
        {
            printf("%d ", col);
        }
        printf("\n");
    }
}
```

... Output:

Enter the value of n: 5

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

02. Pattern 3 type - 01 : 1

1 2
1 2 3
1 2
1

... code:

```
#include <stdio.h>
int main()
{
    int n;
```

```

printf("Enter the value of n: ");
scanf("%d", &n);

for (int row = 1; row <= n; row++)
{
    for (int col = 1; col <= row; col++)
    {
        printf("%d ", col);
    }
    printf("\n");
}

for (int row = n - 1; row >= 1; row++)
{
    for (int col = 1; col <= row; col++)
    {
        printf("%d ", col);
    }
    printf("\n");
}
}

```

... output :

Enter the value of n: 4

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

Q8. Pattern 4 type : 01 - $\begin{array}{c} 1 \\ 1 \ 2 \\ 1 \ 2 \ 3 \end{array}$

... code :

```
#include <stdio.h>
int main () {
```

```

printf("Enter the value of n: ");
scanf("%d", &n);

for (int row = 1; row <= n, row++)
{
    for (int space = n; space > row, space--)
        printf("   ");
    for (int col = 1; col <= row; col++)
        printf("%d ", col);
    printf("\n");
}

```

... output :

Enter the value of n : 4

```

1
1 2
1 2 3
1 2 3 4

```

Q4. Pattern 5 type : 01 - 1 2 3
 1 2
 1

... code :

```

#include <stdio.h>
int main ()
{
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    for (int row = n; row >= 1, row--)
    {
        for (int space = n; space > row, space--)

```

```

    { printf("   ");
    }

    for(int col=1; col <= row; col++)
    {
        printf("%d ", col);
    }
    printf("\n");
}

}

```

... output :

Enter the value of n: 5

```

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

```

55. Pattern 5 type 02 - 1 0 1
 1 0
 1

... code:

```

#include <stdiah>
int main()
{
    int n;
    printf(" Enter the value of n : ");
    scanf("%d", &n);

    for(int row=n; row >= 1; row--)
    {
        for(int space=n; space > row; space--)
        {
            printf("   ");
        }
        for(int col=1; col <= row; col++)
        {
            if(row%2 == 0)

```

```

    printf("0 ");
}
else
{
    printf("1 ");
}
printf("\n");
}

```

... output:

Enter the value of n: 5

```

1 0 1 0 1
1 0 1 0
1 0 1
1 0
1

```

Q6. Pattern type 03 - ABC
 A B
 A

... code:

```

#include <stdio.h>
int main(){
    int n;
    printf("Enter the value of n:");
    scanf("%d", &n);

    for(int row=n; row >= 1; row--)
    {
        for(int space=n; space > row; space--)
        {
            printf("  ");
        }
        for(int col=1; col <= row; col++)

```

```
    printf("%c", col+64);
}
printf("\n");
{
}
```

... output:

Enter the value of n: 4

A B C D
A B C
A B
A

07. Pattern 5 type 04 - * * *
* *
*

... code:

```
#include <stdio.h>
int main (){
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    for (int row = n; row >= 1, row--)
    {
        for (int space = n; space > row; space--)
        {
            printf(" ");
        }
        for (int col = 1; col <= row; col++)
        {
            printf("* ");
        }
        printf("\n");
    }
}
```

... output:

Enter the value of n: 4

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

08. Pattern 5 type 05- 3 3 3

2 2

1

... code:

```
#include <stdio.h>
int main (){
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    for (int row = n; row >= 1; row--)
    {
        for (int space = n; space > row; space--)
        {
            printf("   ");
        }
        for (int col = 1; col <= row; col++)
        {
            printf("%d ", row);
        }
        printf("\n");
    }
}
```

... output:

Enter the value of n: 4

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

09. Pattern 5 type 06 -

1 1 1
0 0
1

... code:

```
#include <stdio.h>
int main()
{
    int n;
    printf(" Enter the number of row : ");
    scanf("%d", &n);

    for (int row=n; row >= 1; row--)
    {
        for (int space=n; space > row; space--)
        {
            printf("   ");
        }

        for (int col=1; col <= row; col++)
        {
            if (row % 2 == 0)
            {
                printf("0 ");
            }
            else
            {
                printf("1 ");
            }
        }

        printf("\n");
    }
}
```

... output:

Enter the number of row : 5

1 1 1 1 1
0 0 0 0
1 1 1
0 0
1

10. Pattern 5 type : 07 -

C C C
B B
A

... code :

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

    for (int row = n; row >= 1, row--)
    {
        for (int space = n; space > row; space--)
            printf("   ");

        for (int col = 1; col <= row; col++)
        {
            printf("%c ", row + 64);
        }
        printf("\n");
    }
}
```

... output :

Enter the number of rows : 4

D D D D
C C C
B B
A

11. Pattern 5 type 08 - # # #

#

... code :

```
# include <stdio.h>
int main()
```

```

int n;
printf("Enter the number of rows : ");
scanf("%d", &n);

for (int row = n; row >= 1; row--)
{
    for (int space = n; space > row; space--)
        printf("   ");
    for (int col = 1; col <= row; col++)
        printf("#   ");
    printf("\n");
}

```

... output :

Enter the number of rows : 4

```

#####
# #
###
#

```

12. Pattern 6 type 01 - 1 2 3

1	2
1	2
1	

... code :

```

#include <stdio.h>
int main()
{
    int n;
    printf("Enter the number of rows : ");

```

```

for (int row = 1; row <= n; row++)
{
    for (int space = n; space > row; space--)
        printf("   ");
    for (int col = 1; col <= row; col++)
        printf("%d ", col);
    printf("\n");
}

for (int row = n - 1; row >= 1; row--)
{
    for (int space = n; space > row; space--)
        printf("   ");
    for (int col = 1; col <= row; col++)
        printf("%d ", col);
    printf("\n");
}

```

... Output:

Enter the number of rows : 4

```

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

```

Conclusion:

This lab report demonstrate the versatility of loops and nested loops in pattern generation. These programs highlight the ability to manipulate rows, columns, spacing and characters to achieve various outputs. Such pattern are not only visually appealing but also enhance logical thinking and coding skills. These techniques form the foundation for complex problem solving in graphical programming.

References:

- C Standard Library Documentation
- github.com
- freeCodeCamp.org
- StackOverflow.com