

Top Pattern Programs in C



Esha Gupta

Associate Senior Executive

Updated on Oct 3, 2023 11:26 IST

Pattern programs in C are coding exercises where specific patterns or shapes are printed using loops and conditional statements.



Pattern programs in C involve printing different patterns on the console, usually made up of stars, spaces, numbers, or other symbols. These programs help beginners familiarize themselves with loops like for, while & nested and understand the concept of pattern logic.

Explore [Online C Programming Courses](#)

In this blog, we will see examples of 10 pattern programs in C & understand how they work :

- Full Pyramid Pattern
- Right Half Pyramid Pattern



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 01-Nov-2023.

- Left Half Pyramid Pattern
- Inverted Right Half Pyramid Pattern
- Inverted Left Half Pyramid Pattern
- Diamond Pattern
- Hollow Full Pyramid Pattern
- Hollow Inverted Full Pyramid Pattern
- Pascal's Triangle Pattern
- Floyd's Triangle Pattern

Let's understand each of these one by one in detail :

1. Full Pyramid Pattern

It is a triangle with its peak at the top centre of the display, and the base spreads out symmetrically on both sides as it descends. It consists of rows and columns where the number of stars in each row starts with 1 at the top and increases by 2 as you move downward.

Pattern :

```

      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****

```

Code :



Copy code

```
// Full Pyramid Pattern of stars using C
#include <stdio.h>

int main()
{
    int rows = 7;

    // first loop to print all rows
    for (int i = 0; i < rows; i++) {

        // inner loop 1 to print white spaces
        for (int j = 0; j < 2 * (rows - i) - 1; j++) {
            printf(" ");
        }

        // inner loop 2 to print star * character
        for (int k = 0; k < 2 * i + 1; k++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```



What are Loops in C?

In C programming, a loop is a control structure that allows a set of instructions to be executed repeatedly based on a specified condition.



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 01-Nov-2023.



What is if statement in C?

The if statement in C is a control structure that evaluates a condition. If the condition is true, it runs the associated code; if false, the code is ignored. It's...[read more](#)



Control Statements in C | Meaning and Types

Control statements in C are used to determine the order in which the instructions within a program are executed. They are of three types: Selection, Iteration & Jump statements.

2. Right Half Pyramid Pattern

The Right Half Pyramid Pattern is a simpler version of the full pyramid. As the name suggests, only the right half (including the centre) of a full pyramid is displayed. It looks like a right-angled triangle with the right angle at the bottom-left.

Pattern :

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

Code :



```
//Right half pyramid pattern of stars using C
#include <stdio.h>

int main()
{
    int rows = 7;

    // first loop for printing rows
    for (int i = 0; i < rows; i++) {

        // second loop for printing character in each rows
        for (int j = 0; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

3. Left Half Pyramid Pattern

The Left Half Pyramid Pattern is the mirror image of the Right Half Pyramid Pattern. Instead of the right angle being at the bottom-left, it is at the bottom right. It looks like a right-angled triangle with the right angle at the bottom right.

Pattern :

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



* * * *

* * * * *

* * * * * *

* * * * * *

Code :



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 01-Nov-2023.

```
// Left half pyramid pattern of stars using C
#include <stdio.h>

int main()
{
    int rows = 7;

    // first loop is for printing the rows
    for (int i = 0; i < rows; i++) {

        // loop for printing leading whitespaces
        for (int j = 0; j < 2 * (rows - i) - 1; j++) {
            printf(" ");
        }

        // loop for printing * character
        for (int k = 0; k <= i; k++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

4. Inverted Right Half Pyramid Pattern

The Inverted Right Half Pyramid Pattern is actually the Right Half Pyramid, but turned upside down. It's a right-angled triangle where the right angle is now at the



top-left.

Pattern :

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

Code :




```
// Inverted right half pyramid of stars using C
#include <stdio.h>

int main()
{
    int rows = 7;

    // first loop to print all rows
    for (int i = 0; i < rows; i++) {

        // first inner loop to print the * in each row
        for (int j = 0; j < rows - i; j++) {
            printf("* ");
        }
        printf("\n");
    }
}
```

5. Inverted Left Half Pyramid Pattern

The Inverted Left Half Pyramid Pattern is the Left Half Pyramid, but inverted. It's the mirror image of the Inverted Right Half Pyramid. It's a right-angled triangle where the right angle is at the top-right.

Pattern :

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



* * *

* *

*

Code :



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 01-Nov-2023.

```
// Inverted left half pyramid of stars using C
#include <stdio.h>

int main()
{
    int rows = 7;

    // first loop for printing all rows
    for (int i = 0; i < rows; i++) {

        // first inner loop for printing white spaces
        for (int j = 0; j < 2 * i; j++) {
            printf(" ");
        }

        // second inner loop for printing star *
        for (int k = 0; k < rows - i; k++) {
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

6. Diamond Pattern

A diamond pattern is essentially a full pyramid pattern on top of an inverted full



The bottom half resembles an inverted full pyramid but without the row with the maximum width (because the top half already displays it).

```

      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * * * *
* * * * * * * * * * * *
* * * * * * * * * * *
* * * * * * * * * *
* * * * * * * * *
* * * * * * * *
* * * * * * *
* * * * * *
* * * * *
* * * *
* * *
*

```

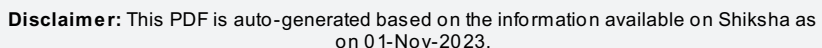
Copy code

```
//Diamond pattern of stars using C
#include <stdio.h>

int main()
{
    int n = 7;

    // first outer loop to iterate through each row
    for (int i = 0; i < 2 * n - 1; i++) {

        // assigning values to the comparator according to
        // the row number
```



```

int comp;
if (i < n) {
    comp = 2 * (n - i) - 1;
}
else {
    comp = 2 * (i - n + 1) + 1;
}

// first inner loop to print leading whitespaces
for (int j = 0; j < comp; j++) {
    printf(" ");
}

// second inner loop to print stars *
for (int k = 0; k < 2 * n - comp; k++) {
    printf("* ");
}
printf("
");
}
return 0;
}

```

7. Hollow Full Pyramid Pattern

The Hollow Full Pyramid Pattern looks like a full pyramid, but its interior is hollow, except for the topmost row and the base. It resembles a triangle with the peak at the top centre and a widening base as you go down.

Pattern :

```

*
* *

```



* *
* *
* *
* *

Code :



Disclaimer: This PDF is auto-generated based on the information available on Shiksha as on 01-Nov-2023.

```
//Hollow full pyramid of stars using C
#include <stdio.h>

int main() {
    int rows = 7;

    // first outer loop to iterate through each row
    for (int i = 0; i < rows; i++) {

        // first inner loop to print leading whitespaces
        for (int j = 0; j < rows - i - 1; j++) {
            printf(" ");
        }

        // second inner loop to print stars * and inner whitespaces
        for (int k = 0; k < 2 * i + 1; k++) {
            if (k == 0 || k == 2 * i) {
                printf("*");
            } else if (i == rows - 1) {
                printf("*");
            } else {
                printf(" ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

8. Hollow Inverted Full Pyramid Pattern

The Hollow Inverted Full Pyramid is actually the upside-down version of the Hollow Full Pyramid. It resembles an upside-down triangle, starting broad at the top and narrowing to a point at the bottom.

Pattern :

* *

* *

* *

* *

* *

*

Code :




```
//Hollow inverted full pyramid of stars using C
#include <stdio.h>

int main() {
    int rows = 7;

    // first loop iterating through each row
    for (int i = 0; i < rows; i++) {

        // first inner loop to print leading white space
        for (int j = 0; j < i; j++) {
            printf(" ");
        }

        // second inner loop to print star* and hollow white space
        for (int k = 0; k < 2 * (rows - i) - 1; k++) {
            if (k == 0 || k == 2 * (rows - i) - 2 || i == 0) {
                printf("*");
            } else {
                printf(" ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

9. Pascal's Triangle Pattern



Pascal's Triangle is a triangular array of numbers. Let's look at a few properties of Pascal's Triangle Pattern below & understand one of the methods we use to print Pascal's Triangle Pattern.

Properties :

- **Edges are 1s:** The numbers on the outer edge of the triangle are all 1s.
- **Row Number:** If you start numbering rows from 0, the n th number in the n th row is n .
- **Binomial Expansion:** The coefficients in the binomial expansion of $(a+b)^n$ are the numbers in the n th row. For example, for $(a+b)^4$, the coefficients are '1 4 6 4 1', corresponding to $a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$.
- **Combinatorial Interpretation:** The entry in the n th row and k th column (starting counting from 0) of Pascal's Triangle gives the number of ways to choose k items from n items, also known as " n choose k " or the binomial coefficient.
- **Hockey Stick Pattern:** If you start from any number and follow a diagonal down and to the right, then turn to follow another diagonal down and to the left, the sum of the numbers in the first diagonal is given by the number where you turn.
- **Fibonacci Sequence:** If you sum the diagonals' numbers starting from the 1s on the outer edges, you get the Fibonacci sequence.

Pattern :

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Example Code :



```
//Pascal's triangle pattern using C
#include <stdio.h>

int main() {
    int rows = 10;

    // outer loop for rows
    for (int i = 0; i < rows; i++) {

        // inner loop 1 for leading white spaces
        for (int j = 0; j < rows - i; j++) {
            printf(" "); // Using two spaces for proper alignment
        }

        int C = 1; // coefficient initialized to 1 for each row's first position

        // inner loop 2 for printing numbers
        for (int k = 0; k <= i; k++) {
            printf("M", C); // Adjusted printing space for the coefficient
            C = C * (i - k) / (k + 1);
        }
        printf("
");
    }
    return 0;
}
```

10. Floyd's Triangle Pattern



In **Floyd's Triangle** Pattern, rather than initiating each row's sequence from the number 1, we consistently print sequential natural numbers across the rows. This pattern can also be used to display a sequence of the alphabet.

Pattern :

```
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 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55
```

Example Code :

[Copy code](#)

```
//Floyd's Triangle Pattern using C
#include <stdio.h>

int main()
{
    int rows = 10;
    int n = 1;

    // Calculate the width of the largest number in Floyd's triangle
    // This is important for maintaining proper alignment in the triangle
    // Especially when we have numbers with more than one digit.

    int width = 0;
```



```

// Calculate the last number in the triangle using the formula  $n(n+1)/2$ 
int lastNumber = rows * (rows + 1) / 2;

// Calculate the number of digits in the last number. This will determine
// the width required for proper alignment.
while (lastNumber) {
    width++;
    lastNumber /= 10;
}

// Outer loop to handle each row of the triangle
for (int i = 0; i < rows; i++) {

    // Inner loop to print numbers in each row.

    // Print each number, using width to ensure proper spacing.
    // "%-*d" is a format specifier where '*' allows dynamic width
    // specification. The '-' ensures left alignment.
    printf("%-*d ", width, n++);
}

    printf("
");
}

return 0;
}

```

Conclusion :

Pattern programs in C are like puzzles that help beginners learn coding basics.



Drawing shapes/patterns with numbers or letters teaches how to use loops and conditions in a fun way. Keep learning, Keep exploring!

What's Next?

Having understood the pattern of programs in C, it's time to build upon that foundation. Here's a suggested roadmap:

- **Data Structures in C:** Familiarize yourself with the concept of data structures and their importance in organizing and storing data efficiently.
- **Pointers in C:** Grasp the fundamental concept of memory addressing and how pointers act as gateways to these addresses.
- **Interview Questions in C:** Prepare for questions on the foundational concepts in C, like data types, operators, and I/O operations.

Boost Your Learning:

Consider enrolling in an online course if you're serious about mastering C. Here are some top recommendations:

- **Coursera's "C for Everyone" Series:** A comprehensive introduction to C, suitable for beginners and those looking to refresh their knowledge.
- **Udemy's "C Programming For Beginners":** This course takes a hands-on approach, ensuring you get practical experience as you learn.

By investing time in a structured course, you'll gain a more profound understanding, benefit from expert insights, and have a clear path of progression.

So, gear up and dive deeper into the world of C programming. The journey you've embarked on is filled with challenges, but the rewards, in terms of knowledge and skills, are immeasurable. Happy coding!

