Example programs for printing patterns in the C programming language.

1. **Right Half Pyramid Pattern**
2. **Left Half Pyramid Pattern**
3. **Full Pyramid Pattern**
4. **Inverted Right Half Pyramid Pattern**
5. **Inverted Left Half Pyramid Pattern**
6. **Inverted Full Pyramid Pattern**
7. **Rhombus Pattern**
8. **Diamond Pattern**
9. **Hourglass Pattern**
10. **Hollow Square Pattern**
11. **Hollow Full Pyramid Pattern**
12. **Hollow Inverted Full Pyramid Pattern**
13. **Hollow Diamond Pattern**
14. **Hollow Hourglass Pattern**
15. **Floyd's Triangle Pattern**
16. **Pascal's Triangle Pattern**

Programs to Print Patterns in C

1. Right Half Pyramid Pattern in C

The right-half pyramid is nothing but a right-angle triangle whose hypotenuse is in the right direction. We can print the right half pyramid pattern using numbers, alphabets, or any other character like a star (*).

```
// C program to print right half pyramid pattern of star
#include <stdio.h>
 int main()
{
   int rows = 5;

   // 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;
}
```

Output

```
*         |  1         |  A

* *       |  1 2       |  A B

* * *     |  1 2 3     |  A B C

* * * *   |  1 2 3 4   |  A B C D

* * * * * |  1 2 3 4 5 |  A B C D E
```

2. Left Half Pyramid Pattern in C

The Left Half Pyramid looks like a right-angled triangle with its hypotenuse facing the left. We can also print this pattern using a character, alphabets, or numbers.

```
// c program to print left half pyramid pattern of star
#include <stdio.h>
int main()
{
   int rows = 5;
   // 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;
}
```

Output

```
     *   |       1   |       A
    * *  |      1 2  |      A B
   * * *  |     1 2 3  |     A B C
  * * * *  |    1 2 3 4  |    A B C D
* * * * *  |   1 2 3 4 5  |   A B C D E
```

3. Full Pyramid Pattern in C

The Full Pyramid pattern looks similar to the Equilateral triangle. We can see this as the combination of the Left Half and Right Half pyramids patterns. The following example demonstrates how to print this pattern using alphabets, numbers, or a star (*).

```
// C program to print the full pyramid pattern of stars
#include <stdio.h>
int main()
{
    int rows = 5;
    // 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;
}
```

Output

```
*          |        1          |        A

* * *       |       1 2 3       |       A B C

* * * * *     |     1 2 3 4 5     |     A B C D E

* * * * * * *   |   1 2 3 4 5 6 7   |   A B C D E F G

* * * * * * * * * | 1 2 3 4 5 6 7 8 9 | A B C D E F G H I
```

4. Inverted Right Half Pyramid Pattern in C

This pattern is the 180° flipped version of the Right Half Pyramid Pattern we discussed earlier.

```
// C program to print the inverted right half pyramid of
// stars
#include <stdio.h>
int main()
{
   int rows = 5;
   // 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");
   } }
```

Output

```
* * * * *  |  1 2 3 4 5  |  A B C D E

* * * *    |  1 2 3 4    |  A B C D

* * *      |  1 2 3      |  A B C

* *        |  1 2        |  A B

*          |  1          |  A
```

5. Inverted Left Half Pyramid Pattern in C

This pattern is the 180° flipped version of the left half pyramid pattern we discussed earlier.

```
// C program to print the inverted left half pyramid pattern
// of stars
#include <stdio.h>
int main()
{
    int rows = 5;
    // 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;
}
```

Output

```
* * * * *   |   1 2 3 4 5   |   A B C D E
 * * * *    |    1 2 3 4    |    A B C D
  * * *     |     1 2 3     |     A B C
   * *      |      1 2      |      A B
    *       |       1       |       A
```

6. Inverted Full Pyramid Pattern in C

It is a 180° flipped version of the Full Pyramid Pattern we discussed earlier. We can see this as the combination of the Inverted Left Half and Inverted Right Half Pyramid Pattern we discussed earlier.

```
// C program to print the inverted full pyramid pattern of
// stars
#include <stdio.h>
int main()
{
    int rows = 5;
    // first loop for printing all rows
    for (int i = 0; i < rows; i++) {
        // first inner loop for printing leading white
        // spaces
        for (int j = 0; j < 2 * i; j++) {
            printf(" ");
        }
        // second inner loop for printing stars *
        for (int k = 0; k < 2 * (rows - i) - 1; k++) {
            printf("* ");
        }
        printf("\n");
    }
}
```

Output

```
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I
 * * * * * * *    |    1 2 3 4 5 6 7    |    A B C D E F G
  * * * * *       |      1 2 3 4 5      |      A B C D E
   * * *          |        1 2 3        |        A B C
    *             |          1          |          A
```

7. Rhombus Pattern in C

The Rhombus pattern is similar to the square pattern, just that we have to add spaces before each line and their count decreases progressively with rows.

```
// C Program to print the rhombus pattern using * star
#include <stdio.h>
int main()
{
   int rows = 5;
   // first outer loop to iterate through each row
   for (int i = 0; i < rows; i++) {
      // first inner loop to print white spaces
      for (int j = 0; j < rows - i - 1; j++) {
         printf(" ");
      }
      // second inner loop to print * star in each row
      for (int k = 0; k < rows; k++) {
         printf("* ");
      }
      printf("\n");
   }
   return 0;
}
```

Output

```
    * * * * *  |     1 2 3 4 5  |     A B C D E
   * * * * *   |    1 2 3 4 5   |    A B C D E
  * * * * *    |   1 2 3 4 5    |   A B C D E
 * * * * *     |  1 2 3 4 5     |  A B C D E
* * * * *      | 1 2 3 4 5      | A B C D E
```

8. Diamond Pattern in C

The Diamond Pattern is obtained by joining the Full Pyramid and Inverted Full Pyramid Pattern by their bases. We can also print this pattern using any character.

```
// C Program to print diamond pattern using star *
#include <stdio.h>
int main()
{
    int n = 5;
    // 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("\n");
    }
    return 0;
}
```

Output

```
      *         |         1         |         A
     * * *      |       1 2 3       |       A B C
   * * * * *    |     1 2 3 4 5     |     A B C D E
 * * * * * * *  |  1 2 3 4 5 6 7    |  A B C D E F G
* * * * * * * * * | 1 2 3 4 5 6 7 8 9 | A B C D E F G H I
 * * * * * * *  |  1 2 3 4 5 6 7    |  A B C D E F G
   * * * * *    |     1 2 3 4 5     |     A B C D E
     * * *      |       1 2 3       |       A B C
      *         |         1         |         A
```

9. Hourglass Pattern in C

Hourglass Pattern is a combination of the inverted full pyramid and full pyramid patterns but in the opposite sense to that of diamond pattern. Here we join them using their tip.

```
// C Program to print hourglass pattern using star *
#include <stdio.h>
int main()
{
   int rows = 5;
   // first outer loop to iterate each row
   for (int i = 0; i < 2 * rows - 1; i++) {
       // assigning comparator
      int comp;
      if (i < rows) {
         comp = 2 * i + 1;
      }
      else {
         comp = 2 * (2 * rows - i) - 3;
      }
      // first inner loop to print leading spaces
      for (int j = 0; j < comp; j++) {
```

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

Output

```
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I
 * * * * * * *    |    1 2 3 4 5 6 7    |    A B C D E F G
  * * * * *       |      1 2 3 4 5      |      A B C D E
   * * *          |        1 2 3        |        A B C
    *             |          1          |          A
   * * *          |        1 2 3        |        A B C
  * * * * *       |      1 2 3 4 5      |      A B C D E
 * * * * * * *    |    1 2 3 4 5 6 7    |    A B C D E F G
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I
```

10. Hollow Square Pattern in C

The Hollow Square Pattern is a square with only the boundary lines. The space inside should be empty in this pattern.

```
// C Program to print hollow square pattern using * star
#include <stdio.h>
int main()
{
    int rows = 5;
    // outer loop to iterator through each row
    for (int i = 0; i < rows; i++) {
```

```
        // inner loop to print * star in each row
        for (int j = 0; j < rows; j++) {
            // statement to check boundry condition
            if (i > 0 && i < rows - 1 && j > 0
                && j < rows - 1) {
                printf("  ");
            }
            else {
                printf("* ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
* * * * *  |  1 2 3 4 5  |  A B C D E

*      *  |  1      5  |  A      E

*      *  |  1      5  |  A      E

*      *  |  1      5  |  A      E

* * * * *  |  1 2 3 4 5  |  A B C D E
```

11. Hollow Full Pyramid Pattern

In the Hollow Pyramid pattern, we only have to print the boundary of the full pyramid.

```
// C program to print hollow full pyramid using star *
#include <stdio.h>
int main()
{
    int rows = 5;
    // first outer loop to iterate through each loop
    for (int i = 0; i < rows; i++) {
```

```
        // first inner loop to print leading whitespaces
        for (int j = 0; j < 2 * (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 || i == rows - 1) {
                printf("* ");
            }
            else {
                printf("  ");
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
      *         |         1         |         A

    *   *       |       1   3       |       A   C

  *       *     |     1       5     |     A       E

*           *   |   1           7   |   A           G

* * * * * * * * * | 1 2 3 4 5 6 7 8 9 | A B C D E F G H I
```

12. Hollow Inverted Full Pyramid Pattern

In this pattern, we print the inverted full pyramid with only boundary elements and remove the inside elements to make it hollow.

```
// C Program to print hollow full pyramid pattern using star
// *
#include <stdio.h>
```

```
int main()
{
    int rows = 5;
    // 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 < 2 * i + 1; 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;
}
```

Output

```
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I

 *       *   |   1       7   |   A       G

  *     *    |    1     5    |    A     E

   *  *      |      1  3      |      A  C

    *        |        1       |        A
```

13. Hollow Diamond Pattern

This pattern is also similar to the Diamond Pattern but without the inner elements such that it appears hollow inside.

```
// C Program to print the hollow diamond pattern using star
// *
#include <stdio.h>
int main()
{
    int n = 5;
    // first outer loop to iterator 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 star * and inner
        // whitespaces
        for (int k = 0; k < 2 * n - comp; k++) {
            if (k == 0 || k == 2 * n - comp - 1) {
                printf("* ");
            }
            else {
                printf("  ");
            }
        }
        printf("\n");
    }
    return 0;
```

}

Output

```
      *       |       *       |       A
    *   *     |     *   *     |     A   C
  *       *   |   *       *   |   A       E
*           * | *           * | A           G
*           * | *           * | A           I
*           * | *           * | A           G
  *       *   |   *       *   |   A       E
    *   *     |     *   *     |     A   C
      *       |       *       |       A
```

14. Hollow Hourglass Pattern in C

The hollow hourglass is the pattern in which only the boundary of the hourglass pattern is visible.

```
// C Program to print the hourglass pattern using star *
#include <stdio.h>
int main()
{
    int n = 5;
    // first outer loop to iterate through each row
    for (int i = 0; i < 2 * n - 1; i++) {
        // assigning comparator
        int comp;
        if (i < n) {
            comp = 2 * i + 1;
        }
        else {
            comp = 2 * (2 * n - i) - 3;
        }
```

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

Output

```
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I
 *       *   |   1       7    |   A       G
  *     *    |    1     5     |    A     E
   * *      |     1 3      |     A C
    *       |      1       |      A
   * *      |     1 3      |     A C
  *     *    |    1     5     |    A     E
 *       *   |   1       7    |   A       G
* * * * * * * * *  |  1 2 3 4 5 6 7 8 9  |  A B C D E F G H I
```

15. Floyd's Triangle in C

In Floyd's Triangle pattern, instead of starting the sequence of the numbers from 1 in each row, we print consecutive natural numbers. We can also print this pattern for alphabet sequence.

```
// C Program to print the Floyd's Triangle
#include <stdio.h>
int main()
{
   int rows = 4;
   int n = 1;
   // outer loop to print all rows
   for (int i = 0; i < rows; i++) {
      // innter loop to print abphabet in each row
      for (int j = 0; j <= i; j++) {
         printf("%d ", n++);
      }
      printf("\n");
   }
   return 0;
}
```

Output

```
1         |  A

2 3       |  B C

4 5 6     |  D E F

7 8 9 10  |  G H I J
```

16. Pascal's Triangle in C

A Pascal's Triangle is a triangular array of binomial coefficients where the n^{th} row contains the binomial coefficients ${}^nC_0$, ${}^nC_1$, ${}^nC_2$, ……. ${}^nC_n$. The following example demonstrates one of the methods using which we can print Pascal's Triangle Pattern.

```
// C program to print the pascal's triangle pattern
#include <stdio.h>
int main()
{
    int rows = 5;
    // outer loop for rows
    for (int i = 1; i <= rows; i++) {
        // inner loop 1 for leading white spaces
        for (int j = 0; j < rows - i; j++) {
            printf(" ");
        }
        int C = 1; // coefficient
          // inner loop 2 for printing numbers
        for (int k = 1; k <= i; k++) {
            printf("%d ", C);
            C = C * (i - k) / k;
        }
        printf("\n");
    }
    return 0;
}
```

Output

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