1. Square Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Size of the square
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Explanation: This program prints a square of asterisks. The outer loop runs n times for each row, and the inner loop runs n times for each column, printing an asterisk followed by a space. After printing all columns in a row, it moves to the next line.

2. Right Triangle Pattern

```
c
Copy code
```

```c
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Output:

markdown

Copy code

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

Explanation: This program creates a right triangle. For each row `i`, it prints `i` asterisks. The outer loop controls the number of rows, and the inner loop controls the number of asterisks printed in each row.

3. Inverted Right Triangle Pattern

c

Copy code

```c
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
* * * * *
* * * *
* * *
* *
*
```

Explanation: This program prints an inverted right triangle. The outer loop runs from n to 1, reducing the number of asterisks printed in each subsequent row.

4. Pyramid Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the pyramid
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            printf("*"); // Print asterisks
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   *
   ***
  *****
 *******
*********
```

Explanation: This program prints a pyramid. The first inner loop prints spaces to center the asterisks, and the second inner loop prints the required number of asterisks for the current row.

5. Inverted Pyramid Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the pyramid
    for (int i = n; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            printf("*"); // Print asterisks
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
*********
 *******
  *****
   ***
    *
```

Explanation: This prints an inverted pyramid. Similar to the pyramid pattern but starts from the widest row and decreases the number of asterisks while increasing the spaces.

6. Diamond Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the diamond
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            printf("*"); // Print asterisks
        }
        printf("\n");
    }
    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            printf("*"); // Print asterisks
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Explanation: This program prints a diamond shape. It first prints an upward pyramid and then an inverted pyramid to complete the diamond shape.

7. Hollow Square Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Size of the square
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i == 0 || i == n - 1 || j == 0 || j == n - 1) {
                printf("* "); // Print border
            } else {
                printf("  "); // Print space
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

markdown
Copy code

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

Explanation: This prints a hollow square. It checks if the current position is on the border (first/last row or column) and prints an asterisk; otherwise, it prints a space.

8. Hollow Right Triangle Pattern

c
Copy code

```
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 0; i < n; i++) {
        for (int j = 0; j <= i; j++) {
            if (j == 0 || i == n - 1 || i == j) {
                printf("* "); // Print border
            } else {
                printf("  "); // Print space
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
*
* *
*   *
*     *
* * * * *
```

Explanation: This prints a hollow right triangle. It prints asterisks on the edges and the base, leaving the inside empty.

9. Number Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", j); // Print numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Explanation: This program prints a number pattern where each row contains numbers from 1 to the row number. The outer loop iterates through the rows, while the inner loop prints numbers.

10. Reverse Number Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", j); // Print numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
```

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

Explanation: This prints a reverse number pattern. The outer loop goes from n to 1, and the inner loop prints numbers from 1 to the current row number.

11. Pascal's Triangle

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 0; i < n; i++) {
        int num = 1; // Initialize the first number of the row
        for (int j = 0; j <= i; j++) {
            printf("%d ", num);
            num = num * (i - j) / (j + 1); // Calculate next number
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
1
1 1
1 2 1
1 3 3 1
```

```
1 4 6 4 1
```

Explanation: This program prints Pascal's triangle. Each number is calculated based on the previous numbers in the row using the formula for combinations.

12. Floyd's Triangle

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    int num = 1;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", num++); // Print and increment num
        }
        printf("\n");
    }
    return 0;
}
```

Output:

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

Explanation: This prints Floyd's triangle. The variable num starts from 1 and increments with each printed number, filling each row with sequential numbers.

13. Diamond Number Pattern

c

Copy code

```
#include <stdio.h>

int main() {
    int n = 5; // Height of the diamond
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("%d ", k); // Print numbers
        }
        printf("\n");
    }
    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("%d ", k); // Print numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

markdown

Copy code

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

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

Explanation: This prints a diamond number pattern. The first loop creates the upper half, while the second loop creates the lower half, centering the numbers in both halves.

14. Hollow Diamond Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the diamond
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            if (k == 1 || k == (2 * i - 1)) {
                printf("*"); // Print border
            } else {
                printf(" "); // Print space
            }
        }
        printf("\n");
    }
    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= (2 * i - 1); k++) {
            if (k == 1 || k == (2 * i - 1)) {
                printf("*"); // Print border
            } else {
                printf(" "); // Print space
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

Explanation: This prints a hollow diamond. The program checks if the current position is at the border and prints an asterisk; otherwise, it prints a space.

15. Right-Aligned Triangle Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("*"); // Print asterisks
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   *
   **
  ***
 ****
*****
```

Explanation: This prints a right-aligned triangle. The outer loop controls the rows, while the first inner loop prints spaces to align the triangle to the right, and the second inner loop prints the asterisks.

16. Character Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 0; i < n; i++) {
        for (int j = 0; j <= i; j++) {
            printf("%c ", 'A' + j); // Print characters
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
css
Copy code
A
A B
A B C
A B C D
A B C D E
```

Explanation: This program prints an alphabet pattern. The outer loop iterates through rows, and the inner loop prints characters starting from 'A', calculated by adding the loop index to the ASCII value of 'A'.

17. Reverse Character Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = n; i >= 1; i--) {
        for (int j = 0; j < i; j++) {
            printf("%c ", 'A' + j); // Print characters
        }
        printf("\n");
    }
    return 0;
}
```

Output:

css
Copy code

```
A B C D E
A B C D
A B C
A B
A
```

Explanation: This prints a reverse character pattern. The outer loop goes from n to 1, and the inner loop prints characters starting from 'A' to the current row length.

18. Increasing Star Pattern

c
Copy code

```c
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("* "); // Print stars
        }
        printf("\n");
    }
    return 0;
}
```

Output:

markdown
Copy code

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

Explanation: This is similar to the right triangle pattern but focuses solely on stars. Each row prints an increasing number of stars based on the row number.

19. Decreasing Star Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            printf("* "); // Print stars
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
* * * * *
* * * *
* * *
* *
*
```

Explanation: This program prints a decreasing star pattern. The outer loop runs from n to 1, and the inner loop prints stars according to the current row index.

20. Zigzag Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the zigzag
    for (int i = 1; i <= n; i++) {
        if (i % 2 == 1) {
            for (int j = 1; j <= n; j++) {
                printf("* "); // Print stars for odd rows
            }
        } else {
            for (int j = 1; j <= n; j++) {
                printf("  "); // Print spaces for even rows
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
* * * * *

* * * * *

* * * * *
```

Explanation: This program prints a zigzag pattern. For odd rows, it prints stars, while for even rows, it prints spaces.

21. Right-Aligned Number Triangle

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("%d ", k); // Print numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

Explanation: This pattern prints numbers in a right-aligned triangle. The outer loop determines the rows, the first inner loop prints spaces, and the second inner loop prints the numbers from 1 to the current row index.

22. Number Pyramid Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the pyramid
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("%d ", k); // Print numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
   1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

Explanation: This program prints a number pyramid. The first inner loop prints spaces to center the numbers, while the second inner loop prints the numbers from 1 to the current row index.

23. Number Pattern with Decrementing Values

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i; j++) {
            printf("%d ", n - i); // Print decremented numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

Explanation: This program prints a pattern where each row has the same number printed, starting from n and decrementing with each row.

24. Alternating Number Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            if (i % 2 == 0) {
                printf("%d ", 2); // Print 2 for even rows
            } else {
                printf("%d ", 1); // Print 1 for odd rows
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
1
2 2
1 1 1
2 2 2 2
1 1 1 1 1
```

Explanation: This program prints alternating numbers based on the row index. Odd rows contain the number 1, and even rows contain the number 2.

25. Multiplication Table Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Size of the multiplication table
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            printf("%d ", i * j); // Print product
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

Explanation: This program prints a multiplication table of size n. The outer loop iterates over the rows, while the inner loop computes and prints the product of the current row and column indices.

26. Increasing Even Number Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", 2 * j); // Print even numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

Copy code

```
2
2 4
2 4 6
2 4 6 8
2 4 6 8 10
```

Explanation: This program prints increasing even numbers. For each row, it prints the first `i` even numbers, which are calculated by multiplying `j` by 2.

27. Decreasing Odd Number Pattern

c

Copy code

```c
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    for (int i = n; i >= 1; i--) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", 2 * j - 1); // Print odd numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

Copy code

```
1 3 5 7 9
1 3 5 7
```

```
1 3 5
1 3
1
```

Explanation: This program prints decreasing odd numbers. The outer loop decreases the row count, while the inner loop prints the first `i` odd numbers.

28. Fibonacci Pattern

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Number of rows
    int a = 0, b = 1, c;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("%d ", a); // Print Fibonacci numbers
            c = a + b; // Next Fibonacci number
            a = b; // Update a
            b = c; // Update b
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Copy code
0
0 1
0 1 1
0 1 1 2
```

```
0 1 1 2 3
```

Explanation: This program prints Fibonacci numbers in a triangular pattern. The first two Fibonacci numbers (0 and 1) are initialized, and each subsequent number is the sum of the two preceding ones.

29. Right-Aligned Even Number Triangle

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Height of the triangle
    for (int i = 1; i <= n; i++) {
        for (int j = n; j > i; j--) {
            printf(" "); // Print leading spaces
        }
        for (int k = 1; k <= i; k++) {
            printf("%d ", 2 * k); // Print even numbers
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
         2
        2 4
      2 4 6
    2 4 6 8
  2 4 6 8 10
```

Explanation: This program prints even numbers in a right-aligned triangle. The outer loop determines the row, the first inner loop prints spaces, and the second inner loop prints the first `i` even numbers.

30. Cross Pattern with Numbers

```
c
Copy code
#include <stdio.h>

int main() {
    int n = 5; // Size of the cross
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (j == i || j == (n - i + 1)) {
                printf("%d ", j); // Print diagonal numbers
            } else {
                printf("  "); // Print space
            }
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
markdown
Copy code
1       5
  2   4
    3
  2   4
1       5
```

Explanation: This program prints a cross pattern with numbers. It checks if the current position is on either diagonal of the cross and prints the row number; otherwise, it prints a space.