1. Left Half Pyramid Pattern

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    int rows = 5;

    // This loop for traverse
    // pyramid from top to bottom
    for (int i = 0; i < rows; i++) {

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

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


=== Code Execution Successful ===
```

main.c — Share — Run — Output

```c
#include <stdio.h>

int main()
{
    int rows = 5;

    // This loop for traverse pyramid from top to bottom
    for (int i = 0; i < rows; i++)
    {

        // Inner loop for printing character in each rows
        for (int j = 0; j <= i; j++)
        {
            printf("%d ", j + 1);
        }
        printf("\n");
    }
    return 0;
}
```

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


=== Code Execution Successful ===
```

2. Right Half Pyramid Pattern

main.c — Share — Run — Output

```c
#include <stdio.h>

int main()
{
    int rows = 5;

    // This loop for traverse pyramid from top to bottom
    for (int i = 0; i < rows; i++)
    {

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

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

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


=== Code Execution Successful ===
```

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    int rows = 5;

    // This loop for traverse
    // pyramid from top to bottom
    for (int i = 0; i < rows; i++) {

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

        // This loop for printing
        // continious numbers in each row
        for (int k = 0; k <= i; k++) {
            printf("%d ", k + 1);
        }
        printf("\n");
    }
    return 0;
}
```

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


=== Code Execution Successful ===
```

3. Inverted Left Half Pyramid Pattern

main.c — Share — Run — Output

```c
#include <stdio.h>

int main()
{
    int rows = 5;

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

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

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


=== Code Execution
```

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    int rows = 5;

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

        // Inner loop to print the
        // numbers in each row
        for (int j = 0; j < rows - i; j++) {
            printf("%d ", j + 1);
        }
        printf("\n");
    }
}
```

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


=== Code Execution
```

4. Inverted Right Half Pyramid Pattern

main.c

Share Run

```c
#include <stdio.h>
int main()
{
    int rows = 5;

    // Outer 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

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

=== Code Execution Successful ==
```

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int rows = 5;

    // Outer 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 numbers
        for (int k = 0; k < rows - i; k++) {
            printf("%d ", k + 1);
        }
        printf("\n");
    }

    return 0;
}
```

Output

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

=== Code Execution Successful ===
```

5. Full Pyramid Pattern

main.c

```c
#include <stdio.h>

int main()
{
    int rows = 5;

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

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

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

Output

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


=== Code Execution Successful ===
```

6. Pascal's Triangle

main.c

```c

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(" ");

        // coefficient
        int C = 1;

        // 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


=== Code Execution Successf
```

7. Floyd's Triangle

main.c

Share Run

```c
#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
2 3
4 5 6
7 8 9 10


=== Code Execution Successful
```

main.c

Share Run

```c
// C Program to print the Floyd's Triangle of Alphabets
#include <stdio.h>

int main()
{
    int rows = 4;
    char n = 'A';

    // 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("%c ", n++);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
A
B C
D E F
G H I J


=== Code Execution Successful ===
```