

1. Left Half Pyramid Pattern

The screenshot shows a code editor interface with a dark theme. On the left, the code file 'main.c' is displayed with syntax highlighting for C code. On the right, the 'Output' panel shows the execution results. The code prints a left half pyramid pattern of asterisks (*). The output consists of five rows of asterisks, increasing in length from one star in the first row to five stars in the fifth row.

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5
6     // This loop for traverse
7     // pyramid from top to bottom
8     for (int i = 0; i < rows; i++) {
9
10        // Inner loop for printing
11        // character in each rows
12        for (int j = 0; j <= i; j++) {
13            printf("* ");
14        }
15        printf("\n");
16    }
17    return 0;
18 }
```

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

== Code Execution Successful ==

The screenshot shows a code editor interface with a dark theme. On the left, the code file 'main.c' is displayed with syntax highlighting for C code. On the right, the 'Output' panel shows the execution results. The code prints a left half pyramid pattern of numbers (1 to 5). The output consists of five rows of numbers, increasing in length from one number in the first row to five numbers in the fifth row.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int rows = 5;
6
7     // This loop for traverse pyramid from top to bottom
8     for (int i = 0; i < rows; i++)
9     {
10
11        // Inner loop for printing character in each rows
12        for (int j = 0; j <= i; j++)
13        {
14            printf("%d ", j + 1);
15        }
16        printf("\n");
17    }
18    return 0;
19 }
```

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				Run	Output
<pre>1 #include <stdio.h> 2 3 int main() 4 { 5 int rows = 5; 6 7 // This loop for traverse pyramid from top to bottom 8 for (int i = 0; i < rows; i++) 9 { 10 11 // This loop for printing leading whitespaces 12 for (int j = 0; j < 2 * (rows - i) - 1; j++) 13 { 14 printf(" "); 15 } 16 17 // This loop for printing * character in each row 18 for (int k = 0; k <= i; k++) 19 { 20 printf("* "); 21 } 22 printf("\n"); 23 } 24 return 0; 25 }</pre>				<pre>* * * * * * * * * * * * * * * ==== Code Execution Successful ===</pre>	

main.c				Run	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int rows = 5; 5 6 // This loop for traverse 7 // pyramid from top to bottom 8 for (int i = 0; i < rows; i++) { 9 10 // This loop for printing 11 // leading whitespaces 12 for (int j = 0; j < 2 * (rows - i) - 2; j++) { 13 printf(" "); 14 } 15 16 // This loop for printing 17 // continious numbers in each row 18 for (int k = 0; k <= i; k++) { 19 printf("%d ", k + 1); 20 } 21 printf("\n"); 22 } 23 return 0; 24 }</pre>				<pre>1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 ==== Code Execution Successful ===</pre>	

3. Inverted Left Half Pyramid Pattern

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing 'main.c' with the following C code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int rows = 5;
6
7     // Outer loop to print all rows
8     for (int i = 0; i < rows; i++)
9     {
10
11         // Inner loop to print the * in each row
12         for (int j = 0; j < rows - i; j++)
13         {
14             printf("* ");
15         }
16         printf("\n");
17     }
18 }
```

On the right is the 'Output' pane which displays the printed output of the code:

```
* * * * *
* * * *
* * *
* *
*
==== Code Execution
```

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing 'main.c' with the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5
6     // Outer loop to print all rows
7     for (int i = 0; i < rows; i++) {
8
9         // Inner loop to print the
10        // numbers in each row
11        for (int j = 0; j < rows - i; j++) {
12            printf("%d ", j + 1);
13        }
14        printf("\n");
15    }
16 }
```

On the right is the 'Output' pane which displays the printed output of the code:

```
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				Run	Output
<pre>1 #include <stdio.h> 2 int main() 3 { 4 int rows = 5; 5 6 // Outer loop for printing all rows 7 for (int i = 0; i < rows; i++) 8 { 9 // First Inner loop for printing white spaces 10 for (int j = 0; j < 2 * i; j++) 11 { 12 printf(" "); 13 } 14 15 // Second inner loop for printing star * 16 for (int k = 0; k < rows - i; k++) 17 { 18 printf("* "); 19 } 20 printf("\n"); 21 } 22 23 return 0; 24 }</pre>				==== Code Execution Successful =====	<pre>* * * * * * * * * * * *</pre>

main.c				Run	Output
<pre>1 #include <stdio.h> 2 3 int main() { 4 int rows = 5; 5 6 // Outer loop for printing all rows 7 for (int i = 0; i < rows; i++) { 8 9 // First Inner loop for 10 // printing white spaces 11 for (int j = 0; j < 2 * i; j++) { 12 printf(" "); 13 } 14 15 // Second inner loop for 16 // printing numbers 17 for (int k = 0; k < rows - i; k++) { 18 printf("%d ", k + 1); 19 } 20 printf("\n"); 21 } 22 23 return 0; 24 }</pre>				==== Code Execution Successful =====	<pre>1 2 3 4 5 1 2 3 4 1 2 3 1 2 1</pre>

5. Full Pyramid Pattern

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing the following C code:

```
main.c
1 #include <stdio.h>
2
3 int main()
4 {
5     int rows = 5;
6
7     // This loop to print all rows
8     for (int i = 0; i < rows; i++)
9     {
10
11         // Inner loop 1 to print white spaces for each row
12         for (int j = 0; j < 2 * (rows - i) - 1; j++)
13         {
14             printf(" ");
15         }
16
17         // Inner loop 2 to print star (*) character for each row
18         for (int k = 0; k < 2 * i + 1; k++)
19         {
20             printf("* ");
21         }
22         printf("\n");
23     }
24     return 0;
25 }
```

On the right is the output pane, which displays the printed pyramid pattern:

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

Below the output, a message says "==== Code Execution Successful ===".

6. Pascal's Triangle

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing the following C code:

```
main.c
2
3 int main() {
4     int rows = 5;
5
6     // Outer loop for rows
7     for (int i = 1; i <= rows; i++) {
8
9         // Inner loop 1 for leading
10        // white spaces
11         for (int j = 0; j < rows - i; j++)
12             printf(" ");
13
14         // coefficient
15         int C = 1;
16
17         // Inner loop 2 for
18         // printing numbers
19         for (int k = 1; k <= i; k++) {
20             printf("%d ", C);
21             C = C * (i - k) / k;
22         }
23         printf("\n");
24     }
25     return 0;
26 }
```

On the right is the output pane, which displays the first five rows of Pascal's Triangle:

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

Below the output, a message says "==== Code Execution Successf".

7. Floyd's Triangle

main.c				Run	Output
1 #include <stdio.h> 2 3 int main() 4 { 5 int rows = 4; 6 int n = 1; 7 8 // outer loop to print all rows 9 for (int i = 0; i < rows; i++) 10 { 11 12 // inner loop to print alphabet in each row 13 for (int j = 0; j <= i; j++) 14 { 15 printf("%d ", n++); 16 } 17 printf("\n"); 18 } 19 return 0; 20 }				1 2 3 4 5 6 7 8 9 10 ==== Code Execution Successful	

main.c				Run	Output
1 // C Program to print the Floyd's Triangle of Alphabets 2 #include <stdio.h> 3 4 int main() 5 { 6 int rows = 4; 7 char n = 'A'; 8 9 // outer loop to print all rows 10 for (int i = 0; i < rows; i++) { 11 12 // inner loop to print alphabet in each row 13 for (int j = 0; j <= i; j++) { 14 printf("%c ", n++); 15 } 16 printf("\n"); 17 } 18 return 0; 19 }				A B C D E F G H I J ==== Code Execution Successful ===	