

1. Left Half Pyramid Pattern

main.c	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 // 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 }</pre>	<pre>* * * * * * * * * * * * * * *</pre> <p>=== Code Execution Successful ===</p>

main.c	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 // 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 }</pre>	<pre>1 1 2 1 2 3 1 2 3 4 1 2 3 4 5</pre> <p>=== Code Execution Successful ===</p>

2. Right Half Pyramid Pattern

main.c	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> * * * * * * * * * * * * * * *</pre> <p>=== Code Execution Successful ===</p>

main.c	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</pre> <p>=== Code Execution Successful ===</p>

3. Inverted Left Half Pyramid Pattern

main.c	Output
<pre>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 }</pre>	<pre>* * * * * * * * * * * * * * * === Code Execution S</pre>

main.c	Output
<pre>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 }</pre>	<pre>1 2 3 4 5 1 2 3 4 1 2 3 1 2 1 === Code Execution S</pre>

4. Inverted Right Half Pyramid Pattern

main.c	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 10 // First Inner loop for printing white spaces 11 for (int j = 0; j < 2 * i; j++) 12 { 13 printf(" "); 14 } 15 16 // Second inner loop for printing star * 17 for (int k = 0; k < rows - i; k++) 18 { 19 printf("* "); 20 } 21 printf("\n"); 22 } 23 24 return 0; 25 }</pre>	<pre>* * * * * * * * * * * * * * *</pre> <p>=== Code Execution Successful ===</p>

main.c	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>	<pre>1 2 3 4 5 1 2 3 4 1 2 3 1 2 1</pre> <p>=== Code Execution Successful ===</p>

5. Full Pyramid Pattern

main.c	Output
<pre>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 }</pre>	<pre> * * * * * * * * * * * * * * * * * * * * * *</pre> <p>=== Code Execution Successful ===</p>

6. Pascal's Triangle

main.c	Output
<pre>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 }</pre>	<pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1</pre> <p>=== Code Execution Successful ===</p>

7. Floyd's Triangle

main.c	Output
<pre>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 abphabet 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 }</pre>	<pre>1 2 3 4 5 6 7 8 9 10 === Code Execution Successful ===</pre>

main.c	Output
<pre>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 abphabet in each row 13 for (int j = 0; j <= i; j++) { 14 printf("%c ", n++); 15 } 16 printf("\n"); 17 } 18 return 0; 19 }</pre>	<pre>A B C D E F G H I J === Code Execution Successful ===</pre>