✅ EASY — E1: Left Triangle Star Pattern

📥 Input

5

📤 Output

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

🧠 Logic (సరళంగా)

i from 1 to n, ప్రతి పంక్తిలో i times '*' print చేయాలి.

📝 Pseudocode

```
read n
for i = 1 to n:
   print '*' repeated i
```

☕ Java Code (LeftTriangle.java)

```
import java.io.*;

public class LeftTriangle {
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
      int n = Integer.parseInt(br.readLine().trim());
      StringBuilder sb = new StringBuilder();
      for (int i = 1; i <= n; i++) {
         for (int j = 0; j < i; j++) sb.append('*');
         sb.append('\n');
      }
      System.out.print(sb.toString());
```

```
    }
}
```

✅ EASY — E2: Right-Aligned Triangle Star Pattern

📥 Input

```
5
```

📤 Output

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

🧠 Logic

ప్రతి line లో ముందుగా (n-i) spaces, తరువాత i '*' ప్రింట్ చేయాలి.

📝 Pseudocode

```
read n
for i = 1 to n:
    print ' ' repeated (n-i)
    print '*' repeated i
```

☕ Java Code (RightTriangle.java)

```
import java.io.*;

public class RightTriangle {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
```

```
        int n = Integer.parseInt(br.readLine().trim());
        StringBuilder sb = new StringBuilder();
        for (int i = 1; i <= n; i++) {
            for (int s = 0; s < n - i; s++) sb.append(' ');
            for (int j = 0; j < i; j++) sb.append('*');
            sb.append('\n');
        }
        System.out.print(sb.toString());
    }
}
```

◈ MODERATE — M1: Pyramid (Centered Odd Stars)

📥 Input

5

(ఇక్కడ n = number of rows; top row has 1 star, next 3, next 5, ...)

📤 Output

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

🧠 Logic

Row i (1-indexed) లో stars = 2*i - 1; left padding spaces = n - i.

📝 Pseudocode

```
read n
for i = 1 to n:
    print ' ' repeated (n-i)
```

```
   print '*' repeated (2*i - 1)
```

☕ Java Code (Pyramid.java)

```
import java.io.*;

public class Pyramid {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine().trim());
        StringBuilder sb = new StringBuilder();
        for (int i = 1; i <= n; i++) {
            for (int s = 0; s < n - i; s++) sb.append(' ');
            for (int k = 0; k < 2 * i - 1; k++) sb.append('*');
            sb.append('\n');
        }
        System.out.print(sb.toString());
    }
}
```

🔷 MODERATE — M2: Floyd’s Triangle (Number Triangle)

📥 Input

```
4
```

📤 Output

```
1
2 3
4 5 6
7 8 9 10
```

🧠 Logic

సంఖ్యలను ఒక counter (start=1) తీసుకొని ప్రతి row కి row-length వరకూ increment చేస్తూ print చేయాలి.

📝 Pseudocode

```
read n
cnt = 1
for i = 1..n:
   for j = 1..i:
      print cnt; cnt++
   newline
```

☕ Java Code (FloydsTriangle.java)

```java
import java.io.*;

public class FloydsTriangle {
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
      int n = Integer.parseInt(br.readLine().trim());
      StringBuilder sb = new StringBuilder();
      int cnt = 1;
      for (int i = 1; i <= n; i++) {
         for (int j = 1; j <= i; j++) {
            sb.append(cnt);
            if (j < i) sb.append(' ');
            cnt++;
         }
         sb.append('\n');
      }
      System.out.print(sb.toString());
   }
}
```

🔥 HARD — H1: Diamond Star Pattern

📥 Input

5

(Here n is number of rows of upper half including middle; diamond total rows = 2*n - 1)

📤 Output (for n=5)

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

🧠 Logic

Diamond = pyramid of n rows + inverted pyramid of (n-1) rows. Use same centered logic.

📝 Pseudocode

```
read n
# upper including middle
for i = 1..n:
   print ' '*(n-i) + '*'*(2*i-1)
# lower
for i = n-1 down to 1:
   print ' '*(n-i) + '*'*(2*i-1)
```

☕ Java Code (DiamondPattern.java)

```
import java.io.*;
```

```
public class DiamondPattern {
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
      int n = Integer.parseInt(br.readLine().trim());
      StringBuilder sb = new StringBuilder();
      // upper (1..n)
      for (int i = 1; i <= n; i++) {
         for (int s = 0; s < n - i; s++) sb.append(' ');
         for (int k = 0; k < 2 * i - 1; k++) sb.append('*');
         sb.append('\n');
      }
      // lower (n-1 .. 1)
      for (int i = n - 1; i >= 1; i--) {
         for (int s = 0; s < n - i; s++) sb.append(' ');
         for (int k = 0; k < 2 * i - 1; k++) sb.append('*');
         sb.append('\n');
      }
      System.out.print(sb.toString());
   }
}
```

🔥 HARD — H2: Number Palindrome Pyramid (Centered)

📥 Input

```
5
```

📤 Output

```
1
   121
  12321
 1234321
123454321
```

🧠 Logic

ప్రతి row i: left spaces = n-i. Then numbers increasing from 1..i, then decreasing from i-1..1. జాగ్రత్తగా number concatenation.

📝 Pseudocode

```
read n
for i = 1..n:
   print ' '*(n-i)
   for j=1..i: print j (no space)
   for j=i-1..1: print j
   newline
```

🍵 Java Code (NumberPalindromePyramid.java)

```java
import java.io.*;

public class NumberPalindromePyramid {
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
      int n = Integer.parseInt(br.readLine().trim());
      StringBuilder sb = new StringBuilder();
      for (int i = 1; i <= n; i++) {
         for (int s = 0; s < n - i; s++) sb.append(' ');
         // increasing
         for (int j = 1; j <= i; j++) sb.append(j);
         // decreasing
         for (int j = i - 1; j >= 1; j--) sb.append(j);
         sb.append('\n');
      }
      System.out.print(sb.toString());
   }
}
```