Pattern-Based Programming Questions (All 35 Questions - Interview Style)

Square, Rectangle, and Triangle Patterns (1–15)

1. **Solid Square Pattern**
   Problem: Print a solid square of stars of size n.
   Input: n = 4
   Output:

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

Program :-

```
import java.util.*;


public class SolidSquare {


	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);

		System.out.println("Enter the n value : ");

		int n = sc.nextInt();


		for (int i = 0; i < n; i++) {

			for (int j = 0; j < n; j++) {

				System.out.print(" *");

			}

			System.out.println();
```

```
        }
    }
}
```

---

2. **Solid Rectangle Pattern**
   Problem: Print a solid rectangle of `m` rows and `n` columns.
   Input: `m = 3, n = 5`
   Output:

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

Program :-

```
import java.util.Scanner;


public class SolidRect {


    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the m value : ");
        int m = sc.nextInt();
        System.out.println("Enter the n value : ");
        int n = sc.nextInt();


        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(" *");
            }
            System.out.println();
        }
```

```
    }
}
```

---

3. **Right-Angled Triangle (Left-Aligned)**
   Problem: Print a left-aligned right-angled triangle.
   Input: n = 5
   Output:

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

Program :-

```
import java.util.Scanner;


public class RightAngledTri {


    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the n value : ");

        int n = sc.nextInt();


        for (int i = 1; i <= n; i++) {

            for (int j = 1; j <= i; j++) {

                System.out.print(" *");

            }

            System.out.println();

        }

    }

}
```

---

4. **Right-Angled Triangle (Right-Aligned)**
   Input: `n = 5`
   Output:

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

Problem :-

```
import java.util.Scanner;


public class RightAliRightAngle {


	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();


		for (int i = 1; i <= n; i++) {
			for(int k=1;k<=n-i;k++) {
				System.out.print("  ");
			}
			for (int j = 1; j <= i; j++) {
				System.out.print(" *");
			}
			System.out.println();
		}
	}
```

}

---

5. **Inverted Triangle (Left-Aligned)**
   Input: `n = 5`
   Output:

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

Program :-

```
import java.util.Scanner;


public class InvertedTri {


	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);

		System.out.println("Enter the n value : ");

		int n = sc.nextInt();


		for (int i = n; i >= 1 ;i--) {

			for (int j = 1; j <= i; j++) {

				System.out.print(" *");

			}

			System.out.println();

		}

	}

}
```

---

6. **Inverted Triangle (Right-Aligned)**
   Input: `n = 5`
   Output:

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

Program :-

```
import java.util.Scanner;


public class InvertedTriRightAlign {


	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();


		for (int i = n; i >= 1; i--) {
			for (int k = 1; k <= n - i; k++) {
				System.out.print("  ");
			}
			for (int j = 1; j <= i; j++) {
				System.out.print(" *");
			}
			System.out.println();
		}
	}
}
```

---

7. **Centered Pyramid Pattern**
   Input: n = 4
   Output:

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

Program :-

```
import java.util.*;


public class Pyramid {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();

		for(int i=1;i<=n;i++) {
			for(int k=1;k<=n-i;k++) {
				System.out.print("  ");
			}
			for(int j=1;j<=2*i-1;j++) {
				System.out.print(" *");
			}
			System.out.println();
		}
	}
}
```

---

8. **Diamond Pattern**
   Input: n = 3
   Output:

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

Program :-

```
import java.util.Scanner;


public class Diamond {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();

		for (int i = 1; i <= n; i++) {
			for (int k = 1; k <= n - i; k++) {
				System.out.print("  ");
			}
			for (int j = 1; j <= 2 * i - 1; j++) {
				System.out.print(" *");
			}
			System.out.println();
		}
		for (int i = n; i >= 1; i--) {
			for (int k = 1; k <= n - i; k++) {
				System.out.print("  ");
			}
```

```
                for (int j = 1; j <= 2 * i - 1; j++) {
                        System.out.print(" *");
                }
                System.out.println();
            }
        }
}
```

---

9. **Butterfly Pattern**
   Input: n = 4
   Output:

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

Program :-

```
import java.util.*;


public class Butterfly {


        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the n value : ");
                int n = sc.nextInt();


                for (int i = 1; i <= n; i++) {
                        for (int j = 1; j <= i; j++) {
                                System.out.print(" *");
                        }
```

```
                for (int k = 1; k <= 2 * (n - i); k++) {
                        System.out.print("  ");
                }
                for (int j = 1; j <= i; j++) {
                        System.out.print(" *");
                }
                System.out.println();
            }
            for (int i = n; i >= 1; i--) {
                for (int j = 1; j <= i; j++) {
                        System.out.print(" *");
                }
                for (int k = 1; k <= 2 * (n - i); k++) {
                        System.out.print("  ");
                }
                for (int j = 1; j <= i; j++) {
                        System.out.print(" *");
                }
                System.out.println();
            }
        }
}
```

---

10. **Left-Aligned Half Diamond**
    Input: n = 4
    Output:

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

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

Program :-

```
import java.util.*;


public class LeftAliHalfDiamond {
	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();

		for (int i = 1; i <= n; i++) {
			for (int j = 1; j <= i; j++)
				System.out.print(" *");
			System.out.println();
		}

		for (int i = n - 1; i >= 1; i--) {
			for (int j = 1; j <= i; j++)
				System.out.print(" *");
			System.out.println();
		}
	}

}
```

---

11. **Right-Aligned Half Diamond**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;


public class RightAliHalfDiamond {
	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);
		System.out.println("Enter the n value : ");
		int n = sc.nextInt();


		for (int i = 1; i <= n; i++) {
			for(int k=1;k<=n-i;k++)
				System.out.print("  ");
			for (int j = 1; j <= i; j++)
				System.out.print(" *");
			System.out.println();
		}


		for (int i = n - 1; i >= 1; i--) {
			for(int k=1;k<=n-i;k++)
				System.out.print("  ");
```

```
                for (int j = 1; j <= i; j++)
                        System.out.print(" *");
                System.out.println();
            }
        }

}
```

---

12. **Sandglass Pattern**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.*;

public class Sandglass {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the n value : ");
        int n = sc.nextInt();

        for (int i = n; i >= 1; i--) {
            for (int k = 1; k <= n - i; k++)
                System.out.print("  ");
```

```
                for (int j = 1; j <= i; j++)
                        System.out.print(" *");
                System.out.println();
            }
            for (int i = 2; i <= n; i++) {
                for (int k = 1; k <= n - i; k++)
                        System.out.print("  ");
                for (int j = 1; j <= i; j++)
                        System.out.print(" *");
                System.out.println();
            }
        }

}
```

---

13. **Increasing Width Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class IncreasewidTri {


        public static void increasingWidthTriangle(int n) {
                for (int i = 1; i <= n; i++) {
```

```
                for (int j = 1; j <= i; j++) {
                        System.out.print("* ");
                }
                System.out.println();
            }
        }

        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter n: ");
            int n = sc.nextInt();

            increasingWidthTriangle(n);

            sc.close();
        }
}
```

---

14. **Decreasing Width Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;
```

```
public class DecreasewidTri {

	public static void decreasingWidthTriangle(int n) {
		for (int i = n; i >= 1; i--) {
			for (int j = 1; j <= i; j++) {
				System.out.print("* ");
			}
			System.out.println();
		}
	}

	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter n: ");
		int n = sc.nextInt();

		decreasingWidthTriangle(n);

		sc.close();
	}
}
```

---

15. **Right-Aligned Hill Pattern**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;

public class RightaliHillPatn {

  public static void rightAlignedHill(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= (n - i); j++) {
        System.out.print("  ");
      }
      for (int j = 1; j <= i; j++) {
        System.out.print("* ");
      }
      System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    rightAlignedHill(n);

    sc.close();
  }
```

}

Hollow Patterns (16–25)

16. **Hollow Square Pattern**
    Problem: Print a hollow square of stars of size n.
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowSquare {


	public static void main(String[] args) {


		Scanner sc = new Scanner(System.in);

		System.out.println("Enter the n value : ");

		int n = sc.nextInt();


		for (int i = 1; i <= n; i++) {

			for (int j = 1; j <= n; j++) {

				if(i==1 || i==n || j==1 || j==n) {

					System.out.print("* ");

				}

				else {

					System.out.print("  ");
```

```
                }
            }
            System.out.println();
        }
    }
}
```

---

17. **Hollow Rectangle Pattern**
    Problem: Print a hollow rectangle of `m` rows and `n` columns.
    Input: `m = 4, n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowRect {


    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the m value : ");
        int m = sc.nextInt();
        System.out.println("Enter the n value : ");
        int n = sc.nextInt();


        for (int i = 1; i <= m; i++) {
```

```
                for (int j = 1; j <= n; j++) {
                        if(i==1 || i==m || j==1 || j==n) {
                                System.out.print("* ");
                        }
                        else {
                                System.out.print("  ");
                        }
                }
                System.out.println();
        }
    }
}
```

---

18. **Hollow Right-Angled Triangle (Left-Aligned)**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowRightTri {


  public static void hollowRightTriangleLeft(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
```

```
            if (j == 1 || j == i || i == n) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowRightTriangleLeft(n);

    sc.close();
  }
}
```

---

19. **Hollow Right-Angled Triangle (Right-Aligned)**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowRightTriRight {


  public static void hollowRightTriangleRight(int n) {
    for (int i = 1; i <= n; i++) {
      for (int s = 1; s <= (n - i); s++) {
        System.out.print("  ");
      }
      for (int j = 1; j <= i; j++) {
        if (j == 1 || j == i || i == n) {
          System.out.print("* ");
        } else {
          System.out.print("  ");
        }
      }
      System.out.println();
    }
  }


  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();
```

```
    hollowRightTriangleRight(n);


    sc.close();
  }
}
```

---

20. **Hollow Inverted Triangle (Left-Aligned)**
    Input: n = 5
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowInvTriLeft {


  public static void hollowInvertedTriangleLeft(int n) {
    for (int i = n; i >= 1; i--) {
      for (int j = 1; j <= i; j++) {
        if (j == 1 || j == i || i == n) {
          System.out.print("* ");
        } else {
          System.out.print("  ");
        }
      }
      System.out.println();
```

```
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowInvertedTriangleLeft(n);

    sc.close();
  }
}
```

---

21. **Hollow Inverted Triangle (Right-Aligned)**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;

public class HollowInvTriRight {

  public static void hollowInvertedTriangleRight(int n) {
    for (int i = n; i >= 1; i--) {
```

```
        for (int s = 1; s <= (n - i); s++) {
            System.out.print("  ");
        }
        for (int j = 1; j <= i; j++) {
            if (j == 1 || j == i || i == n) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowInvertedTriangleRight(n);

    sc.close();
  }
}
```

---

22. **Hollow Pyramid Pattern**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;


public class HollowPyramid {


  public static void hollowPyramid(int n) {
    for (int i = 1; i <= n; i++) {
      for (int s = 1; s <= (n - i); s++) {
        System.out.print("  ");
      }
      for (int j = 1; j <= (2 * i - 1); j++) {
        if (j == 1 || j == (2 * i - 1) || i == n) {
          System.out.print("* ");
        } else {
          System.out.print("  ");
        }
      }
      System.out.println();
    }
  }


  public static void main(String[] args) {
```

```
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowPyramid(n);

    sc.close();
  }
}
```

---

23. **Hollow Diamond Pattern**
    Input: n = 3
    Output:

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

Program :-

```
import java.util.Scanner;

public class HollowDiamond {

  public static void hollowDiamond(int n) {
    for (int i = 1; i <= n; i++) {
      for (int s = 1; s <= (n - i); s++) {
        System.out.print("  ");
      }
      for (int j = 1; j <= (2 * i - 1); j++) {
```

```
            if (j == 1 || j == (2 * i - 1)) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }

    for (int i = n - 1; i >= 1; i--) {
        for (int s = 1; s <= (n - i); s++) {
            System.out.print("  ");
        }
        for (int j = 1; j <= (2 * i - 1); j++) {
            if (j == 1 || j == (2 * i - 1)) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
```

```
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowDiamond(n);

    sc.close();
  }
}
```

---

24. **Hollow Butterfly Pattern**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;

public class HollowButterfly {

  public static void hollowButterfly(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        if (j == 1 || j == i) {
          System.out.print("* ");
        } else {
```

```
            System.out.print("  ");
        }
    }
    for (int s = 1; s <= 2 * (n - i); s++) {
        System.out.print("  ");
    }
    for (int j = 1; j <= i; j++) {
        if (j == 1 || j == i) {
            System.out.print("* ");
        } else {
            System.out.print("  ");
        }
    }
    System.out.println();
}

for (int i = n; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        if (j == 1 || j == i) {
            System.out.print("* ");
        } else {
            System.out.print("  ");
        }
    }
    for (int s = 1; s <= 2 * (n - i); s++) {
        System.out.print("  ");
```

```
        }
        for (int j = 1; j <= i; j++) {
            if (j == 1 || j == i) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowButterfly(n);

    sc.close();
  }
}
```

---

25. **Hollow Hourglass Pattern**
    Input: `n = 5`
    Output:

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

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

Program :-

```
import java.util.Scanner;


public class HollowHourglass {


   public static void hollowHourglass(int n) {
      // Upper half
      for (int i = n; i >= 1; i--) {
         for (int s = 1; s <= n - i; s++) {
            System.out.print("  ");
         }
         for (int j = 1; j <= (2 * i - 1); j++) {
            if (j == 1 || j == (2 * i - 1)) {
               System.out.print("* ");
            } else {
               System.out.print("  ");
            }
         }
         System.out.println();
      }


      for (int i = 2; i <= n; i++) {
         for (int s = 1; s <= n - i; s++) {
```

```
            System.out.print("  ");
        }
        for (int j = 1; j <= (2 * i - 1); j++) {
            if (j == 1 || j == (2 * i - 1)) {
                System.out.print("* ");
            } else {
                System.out.print("  ");
            }
        }
        System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    hollowHourglass(n);

    sc.close();
  }
}
```

Number-Based Patterns (26–35)

26. **Increasing Number Triangle**
    Problem: Print numbers from 1 to n in triangle form.
    Input: n = 5
    Output:

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

Program :-

```
import java.util.Scanner;


public class IncreaseNumTri {


  public static void increasingNumberTriangle(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print(j + " ");
      }
      System.out.println();
    }
  }


  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();


    increasingNumberTriangle(n);
```

```
    sc.close();
  }
}
```

---

27. **Repeating Row Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class RepRowNumTri {


  public static void repeatingRowNumberTriangle(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print(i + " ");
      }
      System.out.println();
    }
  }


  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
```

```
    int n = sc.nextInt();

    repeatingRowNumberTriangle(n);

    sc.close();
  }
}
```

---

28. **Continuous Number Triangle**
    Input: `n = 4`
    Output:

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

Program :-

```
import java.util.Scanner;

public class ContNumTri {

  public static void continuousNumberTriangle(int n) {
    int num = 1;
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print(num + " ");
        num++;
      }
      System.out.println();
    }
```

```
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter n: ");
        int n = sc.nextInt();

        continuousNumberTriangle(n);

        sc.close();
    }
}
```

---

29. **Reverse Row Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;

public class RevRowNumTri {

    public static void reverseRowNumberTriangle(int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = i; j >= 1; j--) {
```

```
            System.out.print(j + " ");
        }
        System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    reverseRowNumberTriangle(n);

    sc.close();
  }
}
```

---

30. **Inverted Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;

public class IncreaseNumTri {
```

```
  public static void increasingNumberTriangle(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print(j + " ");
      }
      System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    increasingNumberTriangle(n);

    sc.close();
  }
}
```

---

31. **Right-Aligned Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;


public class RightAliNumTri {


  public static void rightAlignedNumberTriangle(int n) {
    for (int i = 1; i <= n; i++) {
      for (int s = 1; s <= n - i; s++) {
        System.out.print("  ");
      }
      for (int j = 1; j <= i; j++) {
        System.out.print(j + " ");
      }
      System.out.println();
    }
  }


  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();


    rightAlignedNumberTriangle(n);


    sc.close();
  }
```

}

---

32. **Pyramid Number Pattern**
    Input: n = 4
    Output:

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

Program :-

```
import java.util.Scanner;


public class PyramidNumTri {


   public static void pyramidNumberPattern(int n) {
      for (int i = 1; i <= n; i++) {
         for (int s = 1; s <= n - i; s++) {
            System.out.print("  ");
         }
         for (int j = 1; j <= i; j++) {
            System.out.print(j + " ");
         }
         for (int j = i - 1; j >= 1; j--) {
            System.out.print(j + " ");
         }
         System.out.println();
      }
   }
```

```
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    pyramidNumberPattern(n);

    sc.close();
  }
}
```

---

33. **Even Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;

public class EvenNumTri {

  public static void evenNumberTriangle(int n) {
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print((2 * j) + " ");
      }
```

```
        System.out.println();
      }
    }

    public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter n: ");
      int n = sc.nextInt();

      evenNumberTriangle(n);

      sc.close();
    }
}
```

---

34. **Odd Number Triangle**
    Input: `n = 5`
    Output:

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

Program :-

```
import java.util.Scanner;

public class OddNumTri {

    public static void oddNumberTriangle(int n) {
```

```
    for (int i = 1; i <= n; i++) {
      for (int j = 1; j <= i; j++) {
        System.out.print((2 * j - 1) + " ");
      }
      System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    oddNumberTriangle(n);

    sc.close();
  }
}
```

---

35. **Pascal's Triangle**
    Input: n = 5
    Output:

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

Program :-

```
import java.util.Scanner;
```

```
public class PascalsTri {

  public static void pascalsTriangle(int n) {
    for (int i = 0; i < n; i++) {
      for (int s = 0; s < n - i - 1; s++) {
        System.out.print("  ");
      }
      int num = 1;
      for (int j = 0; j <= i; j++) {
        System.out.print(num + "   ");
        num = num * (i - j) / (j + 1);
      }
      System.out.println();
    }
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    pascalsTriangle(n);

    sc.close();
  }
```

```
}
```