Arrays→

1. Palindrome→

```
package com.practice;
public class practice {
    public static void main(String[] args) {
        String[] words = {"abc","bcd","aba","cfg","madam"};

        for(int i = 0; i < words.length; i++)
        {
            StringBuilder reversed = new StringBuilder(words[i]);
            String res = reversed.reverse().toString();
             if(words[i].contains(res))
             {
                        System.out.println(words[i]);
                        break;
             }
        }
    }
}
```

2. Index of array→

```
package com.practice;
public class practice {
    public static void main(String[] args) {
        String[] words = {"abc", "bcd", "aaaa", "cbc","efgg","ater","hufr","aadde"};

        for(int i = 0; i < words.length; i++) {
            if(words[i].contains("a")) {
                System.out.print(i + " ");
            }
        }
    }
}
```

3.max end 3–

```
public class LargerElement {
    public static int[] setToLarger(int[] nums) {
        int larger = nums[0] > nums[2] ? nums[0] : nums[2];
        nums[0] = larger;
        nums[1] = larger;
        nums[2] = larger;
        return nums;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3};
        int[] arr2 = {5, 2, 9};
        int[] arr3 = {8, 0, -1};

        int[] modified1 = setToLarger(arr1);
        int[] modified2 = setToLarger(arr2);
        int[] modified3 = setToLarger(arr3);

        System.out.println("Original Array: {" + arr1[0] + ", " + arr1[1] + ", " + arr1[2] + "},
Modified Array: {" + modified1[0] + ", " + modified1[1] + ", " + modified1[2] + "}");
        System.out.println("Original Array: {" + arr2[0] + ", " + arr2[1] + ", " + arr2[2] + "},
Modified Array: {" + modified2[0] + ", " + modified2[1] + ", " + modified2[2] + "}");
        System.out.println("Original Array: {" + arr3[0] + ", " + arr3[1] + ", " + arr3[2] + "},
Modified Array: {" + modified3[0] + ", " + modified3[1] + ", " + modified3[2] + "}");
    }
}
```
4.reverse.
```
public class ReverseArray {
    public static int[] reverseArray(int[] nums) {
        int[] reversedArray = new int[nums.length];
        // Copy elements from the original array in reverse order
        for (int i = 0; i < nums.length; i++) {
            reversedArray[i] = nums[nums.length - 1 - i];
        }
        return reversedArray;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3};
        int[] arr2 = {5, 7, 9};
        int[] arr3 = {-1, 0, 8};

        int[] reversed1 = reverseArray(arr1);
        int[] reversed2 = reverseArray(arr2);
        int[] reversed3 = reverseArray(arr3);

        System.out.println("Original Array: {" + arr1[0] + ", " + arr1[1] + ", " + arr1[2] + "},
Reversed: {" + reversed1[0] + ", " + reversed1[1] + ", " + reversed1[2] + "}");
        System.out.println("Original Array: {" + arr2[0] + ", " + arr2[1] + ", " + arr2[2] + "},
Reversed: {" + reversed2[0] + ", " + reversed2[1] + ", " + reversed2[2] + "}");
        System.out.println("Original Array: {" + arr3[0] + ", " + arr3[1] + ", " + arr3[2] + "},
Reversed: {" + reversed3[0] + ", " + reversed3[1] + ", " + reversed3[2] + "}");
    }
}
```
5.rotate
```
public class RotateLeft {
    public static int[] rotateLeft(int[] nums) {
        int[] rotatedArray = new int[nums.length];
        // Shift elements to the left by one position
        for (int i = 0; i < nums.length - 1; i++) {
            rotatedArray[i] = nums[i + 1];
        }
        // Place the first element at the end
        rotatedArray[nums.length - 1] = nums[0];
        return rotatedArray;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3};
        int[] arr2 = {5, 7, 9};
        int[] arr3 = {-1, 0, 8};

        int[] rotated1 = rotateLeft(arr1);
        int[] rotated2 = rotateLeft(arr2);
        int[] rotated3 = rotateLeft(arr3);

        System.out.println("Original Array: {" + arr1[0] + ", " + arr1[1] + ", " + arr1[2] + "},
Rotated Left: {" + rotated1[0] + ", " + rotated1[1] + ", " + rotated1[2] + "}");
        System.out.println("Original Array: {" + arr2[0] + ", " + arr2[1] + ", " + arr2[2] + "},
Rotated Left: {" + rotated2[0] + ", " + rotated2[1] + ", " + rotated2[2] + "}");
        System.out.println("Original Array: {" + arr3[0] + ", " + arr3[1] + ", " + arr3[2] + "},
Rotated Left: {" + rotated3[0] + ", " + rotated3[1] + ", " + rotated3[2] + "}");
    }
}
```

6.sum 3

```
public class ArraySum {
    public static int sumArray(int[] arr) {
        int sum = 0;
        for (int num : arr) {
            sum += num;
        }
        return sum;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3};
        int[] arr2 = {5, 11, 2};
        int[] arr3 = {7, -3, 9};

        System.out.println("Array: {" + arr1[0] + ", " + arr1[1] + ", " + arr1[2] + "}, Sum: " +
sumArray(arr1));
        System.out.println("Array: {" + arr2[0] + ", " + arr2[1] + ", " + arr2[2] + "}, Sum: " +
sumArray(arr2));
        System.out.println("Array: {" + arr3[0] + ", " + arr3[1] + ", " + arr3[2] + "}, Sum: " +
sumArray(arr3));
    }
}
```

7.Common end

```
public class CommonFirstOrLastElement {
    public static boolean commonFirstOrLastElement(int[] a, int[] b) {
        int aLength = a.length;
        int bLength = b.length;

        // Check if both arrays have at least one element
        if (aLength >= 1 && bLength >= 1) {
            // Check if they have the same first element or the same last element
            return a[0] == b[0] || a[aLength - 1] == b[bLength - 1];
        } else {
            // If either array has no elements, they can't have the same first or last element
            return false;
        }
    }

    public static void main(String[] args) {
        int[] a1 = {1, 2, 3};
        int[] b1 = {7, 2, 5};
        int[] a2 = {1, 2, 3};
        int[] b2 = {7, 2, 3};
        int[] a3 = {1, 2, 3};
        int[] b3 = {1, 2, 3, 4};

        System.out.println("Arrays: {" + a1[0] + ", ..., " + a1[a1.length - 1] + "} and {" + b1[0] + ",
..., " + b1[b1.length - 1] + "}, Common first or last element: " +
commonFirstOrLastElement(a1, b1));
        System.out.println("Arrays: {" + a2[0] + ", ..., " + a2[a2.length - 1] + "} and {" + b2[0] + ",
..., " + b2[b2.length - 1] + "}, Common first or last element: " +
commonFirstOrLastElement(a2, b2));
        System.out.println("Arrays: {" + a3[0] + ", ..., " + a3[a3.length - 1] + "} and {" + b3[0] + ",
..., " + b3[b3.length - 1] + "}, Common first or last element: " +
commonFirstOrLastElement(a3, b3));
    }
}
```

8. Pi array

```
public class FirstThreeDigitsOfPi {
    public static int[] firstThreeDigitsOfPi() {
        return new int[]{3, 1, 4};
    }

    public static void main(String[] args) {
        int[] piDigits = firstThreeDigitsOfPi();
        System.out.println("First three digits of Pi: {" + piDigits[0] + ", " + piDigits[1] + ", " +
piDigits[2] + "}");
    }
}
```

9.first last equal

```
public class FirstLastEqual {
   public static boolean isFirstLastEqual(int[] nums) {
      return nums.length >= 1 && nums[0] == nums[nums.length - 1];
   }

   public static void main(String[] args) {
      int[] arr1 = {1, 2, 3, 4, 1};
      int[] arr2 = {1, 2, 3};
      int[] arr3 = {5};
      int[] arr4 = {7, 7, 7, 7};

      System.out.println("Array: {" + arr1[0] + ", ..., " + arr1[arr1.length - 1] + "}, First and last
elements are equal: " + isFirstLastEqual(arr1));
      System.out.println("Array: {" + arr2[0] + ", ..., " + arr2[arr2.length - 1] + "}, First and last
elements are equal: " + isFirstLastEqual(arr2));
      System.out.println("Array: {" + arr3[0] + ", ..., " + arr3[arr3.length - 1] + "}, First and last
elements are equal: " + isFirstLastEqual(arr3));
      System.out.println("Array: {" + arr4[0] + ", ..., " + arr4[arr4.length - 1] + "}, First and last
elements are equal: " + isFirstLastEqual(arr4));
   }
}
```

10. First last 6

```
public class FirstLast6 {
   public static boolean isFirstLast6(int[] nums) {
      return nums[0] == 6 || nums[nums.length - 1] == 6;
   }

   public static void main(String[] args) {
      int[] arr1 = {1, 2, 6};
      int[] arr2 = {6, 1, 2, 3};
      int[] arr3 = {13, 6, 1, 2, 3};

      System.out.println("Array: {" + arr1[0] + ", ..., " + arr1[arr1.length - 1] + "}, Contains 6 at
first or last position: " + isFirstLast6(arr1));
      System.out.println("Array: {" + arr2[0] + ", ..., " + arr2[arr2.length - 1] + "}, Contains 6 at
first or last position: " + isFirstLast6(arr2));
      System.out.println("Array: {" + arr3[0] + ", ..., " + arr3[arr3.length - 1] + "}, Contains 6 at
first or last position: " + isFirstLast6(arr3));
   }
}.
```

Pattern→

1.triangle..

```
public class patterntri {
   public static void main(String[] args) {
      int rows = 5;
      for (int i = 1; i <= rows ; i++) {
         for (int j = 1; j <= i; j++) {
            System.out.print("*");
         }
         System.out.println();
      }
   }
}
```

2.circle—>

```
import java.util.Scanner;
public class circle {
        public static void main(String[] args) {
                int diameter, xCoord, yCoord, rad, point;
                Scanner ab = new Scanner(System.in);
                System.out.print("Enter the Radius Of Solid Circle: ");
                rad = ab.nextInt();
                diameter = 2 * rad;
                for (int row = 0; row <= diameter; row++) {
                        for (int col = 0; col <= diameter; col++) {
                                xCoord = rad - row;
                                yCoord = rad - col;
                        point = xCoord * xCoord + yCoord * yCoord;

                                if (point <= rad * rad + 1) {
                                        System.out.print("* ");
                                } else {
                                        System.out.print("  ");
                                }
                        }
                        System.out.println();
                }

        }

}
```

3.hollow cicle-

```
import java.util.Scanner;
public class hollowcircle {
        public static void main(String[] args) {
                                double distance;
                                int rad;
                                Scanner sc = new Scanner(System.in);
                                System.out.print("Enter radius: ");
```

```
                    rad = sc.nextInt();
                    for (int row = 0; row <= 2 * rad; row++) {
                        for (int col = 0; col <= 2 * rad; col++) {
                            distance = Math.sqrt((row - rad) * (row - rad) +
(col - rad) * (col - rad));
                            if (distance > rad - 0.5 && distance < rad + 0.5)
                                System.out.print("*");
                            else
                                System.out.print(" ");
                        }
                        System.out.println();
                    }
                }
    }
```

4.Pyramid–

```
public class patterntri {
    public static void main(String args[]) {
    for(int i=1;i<=5;i++) {
        for(int l=1;l<=5-i;l++) {
            System.out.print(" ");
        }
        for(int j=1;j<=i*2-1;j++) {
            System.out.print("%");
        }
        System.out.println();
    }
    }
    }
```

5.hollow triangle

```
// Java Program to print
// Hollow triangle pattern
import java.util.*;

public class GeeksForGeeks {
    // Function to demonstrate pattern
    public static void printPattern(int n)
    {
        int i, j, k;

        // outer loop to handle rows
        for (i = 1; i <= n; i++) {

            // inner loop to print spaces.
            for (j = i; j < n; j++) {
                System.out.print(" ");
            }
```

```
                    for (k = 1; k <= (2 * i - 1); k++) {
                        // printing stars.
                        if (k == 1 || i == n || k == (2 * i - 1)) {
                            System.out.print("*");
                        }
                        // printing spaces.
                        else {
                            System.out.print(" ");
                        }
                    }

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

        // Driver Function
        public static void main(String args[])
        {
                int n = 6;
                printPattern(n);
        }
}
```