ARRAYS

1. Binary Search

```java
public class binarySearch {
    public static int binary_search (int nums[], int key) {

        int start = 0;
        int end = nums.length-1; // means n-1
        while (start <= end) {
            int mid = (start + end) / 2;

            if(nums[mid] == key) {  // FOUND
                return mid;
            }
            if (nums[mid] < key) {  // right means 2nd half
                start = mid + 1;
            }
            else {          // left means 1st half
                end = mid-1;
            }
        }
        return -1;
    }
    public static void main(String[] args) {
        int numbers[] = {2, 4, 6, 8, 10, 12, 14};
        int key = 10;
        System.out.println("Key is at index: " + binary_search(numbers, key));
    }
}
```

2. Buy & Sell Stocks

```java
import java.util.Scanner;
public class buyandSellStocks {
    public static int buy_sell_stocks(int prices[]) { // O(n)
        int buyPrices = Integer.MAX_VALUE;
        int maxProfit = 0;

        for (int i = 0; i < prices.length; i++) { // profit
            if (buyPrices < prices[i]) {
                int profit = prices[i] - buyPrices; // today's profit
                maxProfit = Math.max(maxProfit, profit); // Global max profit
            }
            else {
                buyPrices = prices[i];
            }
        }
        return maxProfit;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter no. of Days: ");
        int days = sc.nextInt();
        int prices[] = new int[days];
        System.out.print("Enter Prices: ");
        for (int i = 0; i < prices.length; i++) {
            prices[i] = sc.nextInt();
        }
        System.out.println("Maximum Profit: " + buy_sell_stocks(prices));
    }
}
```

3. Call by Reference

```java
public class byReference {

    public static void update(int marks[], int nonChangable) {
        nonChangable = 10;
        for (int i = 0; i < marks.length; i++) {
            marks[i] = marks[i] + 1;
        }
    }
    public static void main(String[] args) {
       int marks[] = {97, 98, 99};
       int nonChangable = 5;
       update(marks, nonChangable);
       System.out.println(nonChangable);
       for (int i = 0; i < marks.length; i++) {
        System.out.print(marks[i]+" ");
       }
       System.out.println();
    }
}
```

4. Creation of Array

```java
public class creationofArray {
    public static void main(String[] args) {
        int marks[] = new int[50];
        //  OR int[] marks = new int [50];

        //direct input or intialising
        int num[] = {1, 2, 3};
        int morenum[] = {4, 5, 6};
        String fruits[] = {"apple", "mango", "orange"};
    }
}
```

5. Input of an Array

```java
import java.util.Scanner;

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

        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];

        System.out.println("Enter Elements of Array:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
    }
}
```

6. Largest element of an Array

```java
public class largestNumber {

    public static int largest_number(int nums[]) {

        int largest = Integer.MIN_VALUE; // indicates: -infinity
        int smallest = Integer.MAX_VALUE; // indicates: +infinity

        for (int i = 0; i < nums.length; i++) {
            if (largest < nums[i]) {
                largest = nums[i]; // updating the largest value
            }
            if (smallest > nums[i]) {
                smallest = nums[i];
            }
        }
        System.out.println("Smallest number is: " + smallest);
        return largest;
    }

    public static void main(String[] args) {
       int numbers[] =  {-1, 2, 6, 3, 5};
       System.out.println("Largest number is: " + largest_number(numbers));
    }
}
```

7. Linear Search

```java
public class linearSearch {

    public static int linear_search(int numbers[], int key) {

        for (int i = 0; i < numbers.length; i++) {
            if (numbers[i] == key) {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {

        int numbers[] = {2, 4, 6, 8, 10, 12, 14, 16};
        int key = 12;
        int index = linear_search(numbers, key);
        if (index == -1) {
            System.out.println("NOT FOUND !!!");
        } else {
            System.out.println("Key is at index: " + index);
        }
    }
}
```

8. Max Sub-Array Sum (prefix sum)

```java
import java.util.Scanner;

public class maxSubArray_prefixSum {

    public static void max_subarray_sum (int numbers[]) {
        int currSum = 0;
        int maxSum = Integer.MIN_VALUE;
        int prefix[] = new int[numbers.length];

        prefix [0] = numbers [0];
        // Calculate prefix array
        for (int i = 1; i < prefix.length; i++) {
            prefix[i] = prefix[i-1] + numbers[i];
        }

        for (int i = 0; i < numbers.length; i++) {
            int start = i;
            for (int j = i; j < numbers.length; j++) {
                int end = j;
                currSum = 0;
                currSum = start == 0 ? prefix[end] : prefix[end] - prefix[start-1];
                System.out.print(currSum + " ");
                if (maxSum < currSum) {
                    maxSum = currSum;
                }
            }
        }
        System.out.println("Max Sum = " + maxSum);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Size: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Max SubArray Sum: ");
        max_subarray_sum(arr);
    }
}
```

9. Max Sub-Array Sum

```java
import java.util.Scanner;

public class maxSubarraySum {

    public static void max_subarray_sum (int numbers[]) {
        int currSum = 0;
        int maxSum = Integer.MIN_VALUE;

        for (int i = 0; i < numbers.length; i++) {
            int start = i;
            for (int j = i; j < numbers.length; j++) {
                int end = j;
                currSum = 0;
                for (int k = start; k <= end; k++) {
                    currSum += numbers[k];
                }
                System.out.print(currSum + " ");
                if (maxSum < currSum) {
                    maxSum = currSum;
                }
            }
        }
        System.out.println("Max Sum = " + maxSum);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Size: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Max SubArray Sum: ");
        max_subarray_sum(arr);
    }
}
```

10. Output of an Array

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

        // Creation of an Array
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        // Input of an Array
        System.out.println("Enter the Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        // Output of an array
        for (int i = 0; i < arr.length; i++) {
            System.out.println("Element at Index-" +i+ ":" +arr[i]);
        }
    }
}
```

11. Max Sub-Array Sum (kadane's algorithm)

```java
import java.util.Scanner;
public class maxSubArraySumKADANEalgo {

    public static void kadanes(int numbers[]) {

        int maxSum = Integer.MIN_VALUE;
        int currSum = 0;
        for (int i = 0; i < numbers.length; i++) {
            currSum = currSum + numbers[i];
            if (currSum < 0) {
                currSum = 0;
            }
            maxSum = Math.max(currSum, maxSum);
        }
        System.out.print("Our Max Sub-Array Sum is: " + maxSum);

        // If all numbers are negative then use the following code:
        /*
        int max_so_far = Integer.MIN_VALUE;
        int max_ending_here = 0;
        int max_element = Integer.MIN_VALUE;

        for (int i = 0; i < numbers.length; i++) {
            max_ending_here = Math.max(max_ending_here + numbers[i], 0);
            max_so_far = Math.max(max_ending_here, max_so_far);
            max_element = Math.max(max_element, numbers[i]);
        }
        if (max_so_far == 0) {
            max_so_far = max_element;
        }
        System.out.println(max_so_far);
        */
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Size: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements:");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("By Kadane's Algorithm");
        // System.out.print("Max Sub-Array Sum: ");
        kadanes(arr);
    }
}
```

12. Pairs in an Array

```java
import java.util.Scanner;

public class pairsInArrays {

    public static void pairs_arrays (int numbers[]) {
        int total_pairs = 0;  // Total pairs - n(n-1)/2
        for (int i = 0; i < numbers.length; i++) {
            int curr = numbers[i];
            for (int j = i+1; j < numbers.length; j++) {
                System.out.print("(" +curr+ "," +numbers[j]+ ")");
                total_pairs++;
            }
            System.out.println();
        }
        System.out.println("Total Pairs: " +total_pairs);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Pairs of Given Array:");
        pairs_arrays(arr);
    }
}
```

13. Reverse of an Array

```java
import java.util.Scanner;

public class pairsInArrays {
    public static void pairs_arrays (int numbers[]) {
        int total_pairs = 0;  // Total pairs - n(n-1)/2
        for (int i = 0; i < numbers.length; i++) {
            int curr = numbers[i];
            for (int j = i+1; j < numbers.length; j++) {
                System.out.print("(" +curr+ "," +numbers[j]+ ")");
                total_pairs++;
            }
            System.out.println();
        }
        System.out.println("Total Pairs: " +total_pairs);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter size of Array: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Pairs of Given Array:");
        pairs_arrays(arr);
    }
}
```

14. Linear Search of Strings

```java
public class stringLinearSearch {

    public static int string_linear_search (String str[], String key) {
        for (int i = 0; i < str.length; i++) {
            if (str[i] == key) {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        String str[] = {"Dosa", "Chole Bhatoore", "Chicken", "Fish", "Mutton", "Kabab Paratha", "Samosa"};
        String key = "Fish";
        int index = string_linear_search(str, key);
        if (index == -1) {
            System.out.println("NOT FOUND !!!");
        } else {
            System.out.println("Key is at the index: " + index);
        }
    }
}
```

15. Sub-Arrays

```java
import java.util.Scanner;
public class subArrays {
    public static void sub_arrays (int numbers[]) {
        int sa = 0;  // Total Sub-Arrays = n(n+1)/2
        for (int i = 0; i < numbers.length; i++) {
            int start = i;
            for (int j = i; j < numbers.length; j++) {
                int end = j;
                for (int k = start; k <= end; k++) {
                    System.out.print(numbers[k] + " ");
                }
                sa ++;
                System.out.println();
            }
            System.out.println();
        }
        System.out.println("Total Sub-Arrays: " +sa);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Size: ");
        int size = sc.nextInt();
        int arr[] = new int[size];
        System.out.println("Enter Elements: ");
        for (int i = 0; i < size; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("SubArrays are:");
        sub_arrays(arr);
    }
}
```

16. Trapped Rainwater

```java
import java.util.Scanner;

public class trappedRainwater {

    public static int trapped_rainwater (int height[], int width) { // O(n)
        int n = height.length;

        // Calculate left-max-boundary using auxiliary array
        int leftMax[] = new int[n];
        leftMax[0] = height[0];
        for (int i = 1; i < n; i++) {
            leftMax[i] = Math.max(height[i], leftMax[i-1]);
        }

        // Calculate right-max-boundary using auxiliary array
        int rightMax[] = new int[n];
        rightMax[n-1] = height[n-1];
        for (int i = n-2; i >= 0; i--) {
            rightMax[i] = Math.max(height[i], rightMax[i+1]);
        }

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

            // Waterlevel = min (left-max-boundary, right-max-boundary)
           int waterlevel = Math.min(leftMax[i], rightMax[i]);

            // Trapped Water = (waterlevel - height[i]) * width
            // width = 1
           trappedWater += (waterlevel - height[i]) * width;
        }
        return trappedWater;
    }

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

        System.out.print("Enter Width of the Bars: ");
        int width = sc.nextInt();
        System.out.print("Enter no. of Bars: ");
        int size = sc.nextInt();
        int height[] = new int[size];
        System.out.print("Enter Heights: ");
        for (int i = 0; i < height.length; i++) {
            height[i] = sc.nextInt();
        }
        System.out.println("Total Trapped Water is: " + trapped_rainwater(height, width));
    }
}
```

17. Update of an Array

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

        // DIRECT ASSIGNMENT
        int nums[] = {1, 2, 3, 4, 5, 6};
        nums[2] = 10;
        System.out.println("Updated array elements:");
        for (int num : nums) {
            System.out.print(num + " ");
        }
        System.out.println();

        // USING LOOPS
        int arr[] = {1, 2, 3, 4, 5, 6};
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] > 3) {
                arr[i] = 0;
            }
        }
        System.out.println("Updated array elements:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```