

1. Selection sort

CODE:

```
public class SelectionSort {  
    public static void main(String[] args) {  
        int[] arr = {5, 3, 8, 4, 2};  
        selectionSort(arr);  
        for (int num : arr) System.out.print(num + " ");  
    }  
  
    private static void selectionSort(int[] arr) {  
        for (int i = 0; i < arr.length; i++) {  
            int minIndex = i;  
            for (int j = i + 1; j < arr.length; j++)  
                if (arr[j] < arr[minIndex]) minIndex = j;  
            int temp = arr[i];  
            arr[i] = arr[minIndex];  
            arr[minIndex] = temp;  
        }  
    }  
}
```

OUTPUT:

Output

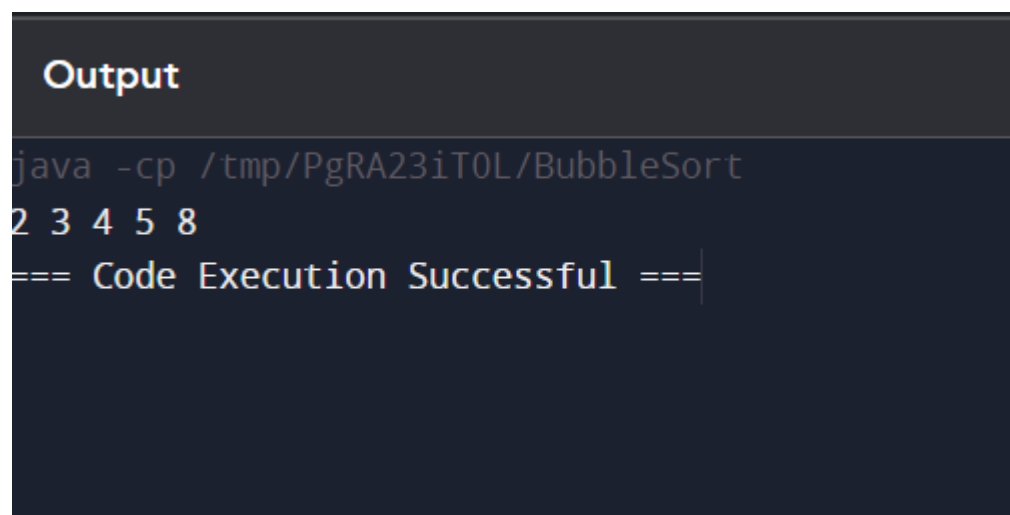
```
java -cp /tmp/Dx67cpgPMB/SelectionSort  
2 3 4 5 8  
=== Code Execution Successful ===
```

2. Bubble sort

CODE:

```
public class BubbleSort {  
    public static void main(String[] args) {  
        int[] arr = {5, 3, 8, 4, 2};  
        bubbleSort(arr);  
        for (int num : arr) System.out.print(num + " ");  
    }  
    private static void bubbleSort(int[] arr) {  
        for (int i = 0; i < arr.length - 1; i++)  
            for (int j = 0; j < arr.length - 1 - i; j++)  
                if (arr[j] > arr[j + 1]) {  
                    int temp = arr[j];  
                    arr[j] = arr[j + 1];  
                    arr[j + 1] = temp;  
                }  
    }  
}
```

OUTPUT:



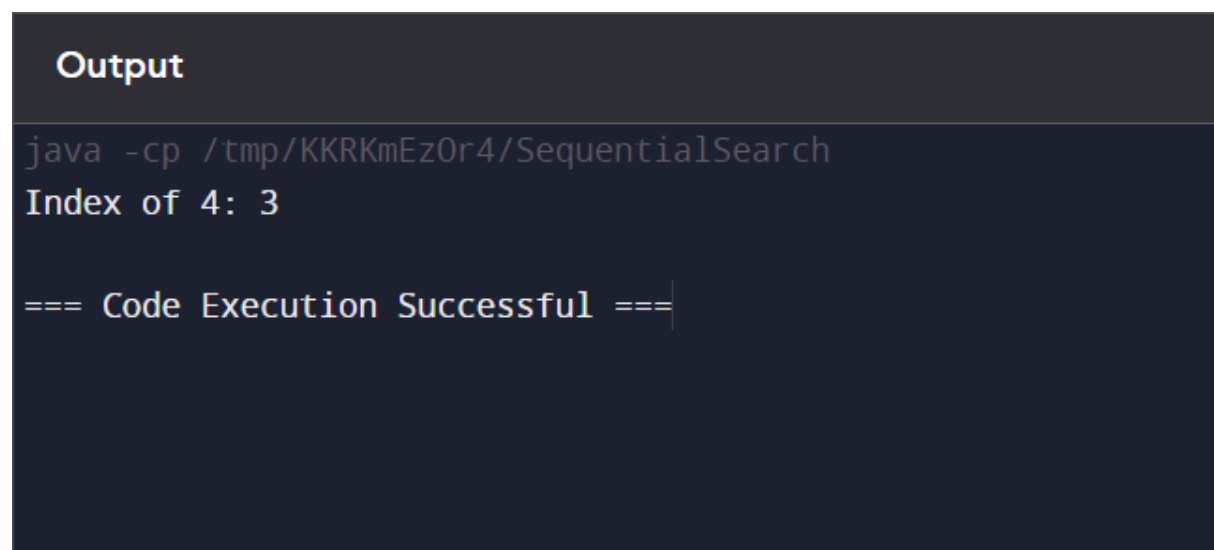
```
Output  
java -cp /tmp/PgRA23iT0L/BubbleSort  
2 3 4 5 8  
=== Code Execution Successful ===
```

3. Linear search

CODE:

```
public class SequentialSearch {  
    public static void main(String[] args) {  
        int[] arr = {5, 3, 8, 4, 2};  
        int target = 4;  
        int index = sequentialSearch(arr, target);  
        System.out.println("Index of " + target + ": " + index);  
    }  
    private static int sequentialSearch(int[] arr, int target) {  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] == target) return i;  
        }  
        return -1;  
    }  
}
```

OUTPUT:

A screenshot of a code execution environment. At the top, there is a dark header with the word "Output" in white. Below the header, the command "java -cp /tmp/KKRKmEz0r4/SequentialSearch" is shown in a light blue font. The output of the program is "Index of 4: 3", displayed in a light blue font. At the bottom, a green message "=== Code Execution Successful ===" is shown in a light blue font.

```
Output  
java -cp /tmp/KKRKmEz0r4/SequentialSearch  
Index of 4: 3  
=== Code Execution Successful ===
```

4. Binary search

CODE:

```
public class BinarySearch {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
        int target = 5;  
        int index = binarySearch(arr, target);  
        System.out.println("Index of " + target + ": " + index);  
    }  
  
    private static int binarySearch(int[] arr, int target) {  
        int left = 0, right = arr.length - 1;  
        while (left <= right) {  
            int mid = left + (right - left) / 2;  
            if (arr[mid] == target) return mid;  
            if (arr[mid] < target) left = mid + 1;  
            else right = mid - 1;  
        }  
        return -1; // Target not found  
    }  
}
```

OUTPUT:

| Output | Clear |
|---|-------|
| java -cp /tmp/nmpSQ14NNa/BinarySearch Index of 5: 4 === Code Execution Successful === | |

