Q1: Bubble Sort in Descending Order

Here's how you can implement Bubble Sort to sort an array in descending order:

Eg:-

```
public class BubbleSortDescending {

    public static void bubbleSortDescending(int[] array) {
        int n = array.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - 1 - i; j++) {
                if (array[j] < array[j + 1]) {
                    // Swap array[j] and array[j + 1]
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] array = {3, 5, 1, 6, 0};
        bubbleSortDescending(array);
        System.out.println("Sorted Array: ");
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
}
```

Eg:-

Q2: Selection Sort in Descending Order

Here's how you can implement Selection Sort to sort an array in descending order:

Eg:-

```
public class SelectionSortDescending {

    public static void selectionSortDescending(int[] array) {
        int n = array.length;
        for (int i = 0; i < n - 1; i++) {
            int maxIndex = i;
            for (int j = i + 1; j < n; j++) {
```

```
            if (array[j] > array[maxIndex]) {
                maxIndex = j;
            }
        }
        // Swap array[i] and array[maxIndex]
        int temp = array[i];
        array[i] = array[maxIndex];
        array[maxIndex] = temp;
    }
  }

  public static void main(String[] args) {
    int[] array = {3, 5, 1, 6, 0};
    selectionSortDescending(array);
    System.out.println("Sorted Array: ");
    for (int num : array) {
        System.out.print(num + " ");
    }
  }
}
```

Eg:-

Q3: Insertion Sort in Descending Order

Here's how you can implement Insertion Sort to sort an array in descending order:

Eg:-

```
public class InsertionSortDescending {

  public static void insertionSortDescending(int[] array) {
    int n = array.length;
    for (int i = 1; i < n; i++) {
        int key = array[i];
        int j = i - 1;
        // Move elements of array[0..i-1], that are less than key, to one position ahead
        while (j >= 0 && array[j] < key) {
            array[j + 1] = array[j];
            j--;
        }
        array[j + 1] = key;
    }
  }

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

```
        int[] array = {3, 5, 1, 6, 0};
        insertionSortDescending(array);
        System.out.println("Sorted Array: ");
        for (int num : array) {
            System.out.print(num + " ");
        }
    }
}
```
Eg:-

Q4: Number of Passes Required for Bubble Sort

To sort an array using Bubble Sort, the number of passes required in the worst case is equal to the number of elements minus one. For an array of size (n), the maximum number of passes needed is (n-1).

Input Array: {3, 5, 1, 6, 0}

- Size of Array: 5
- Number of Passes Required: 5 - 1 = 4

So, it will take up to 4 passes to sort this array in decreasing order.

Q5: Number of Iterations for Selection Sort

Selection Sort always requires \(n-1 \) iterations of the outer loop, where \(n \) is the number of elements in the array. Each iteration involves finding the maximum element in the unsorted part of the array.

Input Array: {3, 5, 1, 6, 0}

- Size of Array: 5
- Number of Iterations: 5 - 1 = 4

Thus, Selection Sort will perform 4 iterations to sort the array in descending order.