1. https://pastebin.com/pVLzyKtW

2. https://pastebin.com/8mSAiQEm

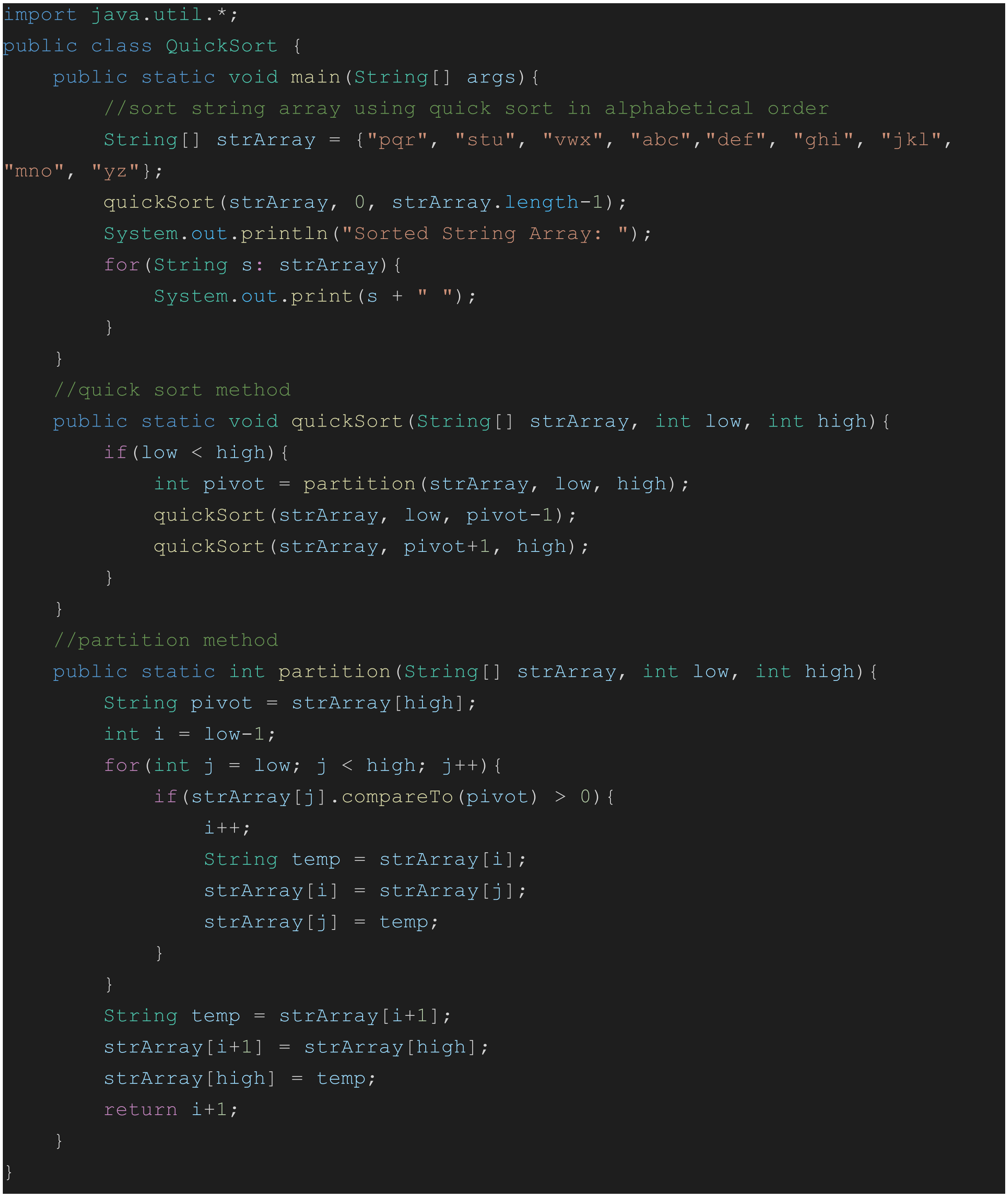
3. Solution:

```
public class Quick {
        public static int partition(int arr[], int low, int high) {
                int pivot = arr[high];
                int i = low - 1;

                for (int j = low; j <= high - 1; j++) {
                        if (arr[j] < pivot) {
                                i++;
                                int temp = arr[i];
                                arr[i] = arr[j];
                                arr[j] = temp;
                        }
                }
                int temp = arr[i + 1];
                arr[i + 1] = arr[high];
                arr[high] = temp;

                for (int e  = 0; e < arr.length; e++) {
                        System.out.print(" " + arr[e]);
                }
                System.out.println("     Pivot: " + pivot);

                return i + 1;
        }

        public static void quickSort(int[] arr, int low, int high) {
                if (low < high) {
                        int pi = partition(arr, low, high);

                        quickSort(arr, low, pi - 1);
                        quickSort(arr, pi + 1, high);
                }
        }

        public static void main(String[] args) {
                int[] arr = {3, 5, 7, 56, 23};


                quickSort(arr, 0, arr.length - 1);
        }
}
```

4. Solution:

```
import java.io.*;
import java.util.Scanner;

public class quickSortFromFile {
        public static int partition(String[] names, int low, int high){
                String pivot = names[high];
        int i = low-1;

        for(int j=low; j<=high-1; j++){
            if(names[j].compareTo(pivot)<0){
                i++;
                String temp = names[i];
                names[i] = names[j];
                names[j]= temp;
            }
        }
        String temp = names[i+1];
        names[i+1] = names[high];
        names[high] = temp;

        return i + 1;
    }
    public static void quickSort(String[] names, int low, int high) {
        if(low<high){
            int pi = partition(names, low, high);

            quickSort(names, low, pi-1);
            quickSort(names, pi+1, high);
        }
    }

    public static void main(String[] args) throws Exception{
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the array size: ");
        int size = input.nextInt();
        System.out.println("Enter the file name: ");
        String fileName = input.next();

        File file = new File(fileName);
        Scanner read = new Scanner(file);

        String[] names = new String[size];

        for(int i=0; i<size; i++){
            names[i] = read.next();
        }
        input.close();
        read.close();

        quickSort(names, 0, names.length-1);

        for (int i=0; i<names.length; i++){
            System.out.println(names[i]);
        }
    }
}
```

5. Solution:



```java
import java.util.*;
public class QuickSort {
    public static void main(String[] args){
        //sort string array using quick sort in alphabetical order
        String[] strArray = {"pqr", "stu", "vwx", "abc","def", "ghi", "jkl",
"mno", "yz"};
        quickSort(strArray, 0, strArray.length-1);
        System.out.println("Sorted String Array: ");
        for(String s: strArray){
            System.out.print(s + " ");
        }
    }
    //quick sort method
    public static void quickSort(String[] strArray, int low, int high){
        if(low < high){
            int pivot = partition(strArray, low, high);
            quickSort(strArray, low, pivot-1);
            quickSort(strArray, pivot+1, high);
        }
    }
    //partition method
    public static int partition(String[] strArray, int low, int high){
        String pivot = strArray[high];
        int i = low-1;
        for(int j = low; j < high; j++){
            if(strArray[j].compareTo(pivot) > 0){
                i++;
                String temp = strArray[i];
                strArray[i] = strArray[j];
                strArray[j] = temp;
            }
        }
        String temp = strArray[i+1];
        strArray[i+1] = strArray[high];
        strArray[high] = temp;
        return i+1;
    }
}
```