# Class ArrayRecursor

```
import java.util.Scanner;
public class ArrayRecursor {
      public static void fill(int arr[]) {
            Scanner s = new Scanner(System.in);
            fill(arr, 0, s);
      }
      private static void fill(int arr[], int start, Scanner input) {
            if(start == arr.length) return;
            System.out.print("Enter number " + (start+1) + ": ");
            int num = input.nextInt();
            arr[start] = num;
            fill(arr, start+1, input);
      }
      public static void print(int arr[]) {
            System.out.print("[");
            print(arr, 0);
            System.out.print("]");
      }
      private static void print(int arr[], int start) {
            System.out.print(arr[start]);
            if (start + 1 != arr.length){
                  System.out.print(", ");
                  print(arr, start+1);
            }
      }

      public static void reverseArray(int arr[]) {
            reverseArray(arr, 0, arr.length-1);
      }
      private static void reverseArray(int arr[], int start, int end) {
            if(start >= end) return;
            //swap
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            reverseArray(arr, start+1, end-1);
      }

      public static int countOcc(int arr[], int x) {
            return countOcc(arr, x, 0);
      }
      private static int countOcc(int arr[], int x, int start) {
            if(start >= arr.length) return 0;
            if(arr[start] == x) return 1 + countOcc(arr,x,start+1);
            return countOcc(arr,x,start+1);
      }
```

```
    public static boolean isPalindrome(int arr[]) {
        return isPalindrome(arr, 0, arr.length-1);
    }
    private static boolean isPalindrome(int arr[], int start, int end) {
        if(start >= end) return true;
        if(arr[start] == arr[end])
            return isPalindrome(arr, start+1, end-1);
        return false;
    }

    public static boolean isSorted(int arr[]) {
        return isSorted(arr, 0);
    }
    private static boolean isSorted(int arr[], int start) {
        if(start == arr.length-1) return true;
        if(arr[start] < arr[start+1]) return isSorted(arr, start+1);
        return false;
    }

    public static int[] mergeTwo(int a[], int b[]) {
        if(!isSorted(a) || !isSorted(b))
            return new int[0];
        int c[] = new int[a.length + b.length];
        mergeTwo(a,b,c,0,0,0);
        return c;
    }
    private static void mergeTwo(int a[], int b[], int c[],
int countA, int countB, int countC) {
        if(countA >= a.length) {
            if(countB >= b.length) return;
            c[countC++] = b[countB++];
        } else if(countB >= b.length || a[countA] < b[countB]){
            c[countC++] = a[countA++];
        }else c[countC++] = b[countB++];
        mergeTwo(a,b,c,countA,countB,countC);
    }

    public static int binarySearch(int arr[], int x) {
        if(!isSorted(arr)) return -2;
        return binarySearch(arr, x, 0, arr.length-1);
    }
    private static int binarySearch(int arr[], int x, int start, int end) {
        if(start > end) return -1;
        int mid = (start + end) / 2;
        if(arr[mid] == x) return mid;
        if(arr[mid] < x) return binarySearch(arr,x,mid+1, end);
        return binarySearch(arr,x,start,mid-1);
    }
```

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int array[] = new int [10];
    for(int i = 0; i < array.length; i++)
        array[i] = -1;
    int choice = 0;
    do{
        System.out.println("1) Enter a new array.");
        System.out.println("2) Print the current array.");
        System.out.println("3) Reverse current array.");
        System.out.println("4) Count occurances.");
        System.out.println("5) Check if array is palindrome");
        System.out.println("6) Merge with another sorted array.");
        System.out.println("7) Search in the sorted array.");
        System.out.println("8) Quit.");
        System.out.print("Enter a choice: ");
        choice = input.nextInt();
        switch(choice){
        case 1:
            fill(array);
            break;
        case 2:
            print(array);
            System.out.println();
            break;
        case 3:
            reverseArray(array);
            print(array);
            System.out.println();
            break;
        case 4:
            System.out.print("Enter x: ");
            int x = input.nextInt();
            int occ = countOcc(array, x);
            System.out.print("The number " + x + " has occured "
                              + occ + " times in ");
            print(array);
            System.out.println();
            break;
        case 5:
            System.out.print("The array ");
            print(array);
            if(isPalindrome(array))
                System.out.println(" is Palindrome");
            else
                System.out.println(" is not Palindrome");
            break;
            case 6:
                System.out.print("Enter the new array size: ");
                int size = input.nextInt();
                int b[] = new int[size];
                System.out.println("Enter the array numbers: ");
                fill(b);
                int c[] = mergeTwo(array,b);
                if(c.length == 0)
            System.out.println("one of the two arrays is not sorted");
                else{
                    System.out.print("Merged: ");
                    print(c);
                    System.out.println();
                }
                break;
            case 7:
                System.out.println("Enter number: ");
                int num = input.nextInt();
                int index = binarySearch(array,num);
                if(index == -1)
            System.out.println("The number " + num + " is not found");
                else
                    System.out.println("The number " + num
                            + " is at index " + index);
                break;
            }
        }while(choice != 8);
        System.out.println("Bye!");
    }
}
```