

Class ArrayRecursor

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

    public static int fill(int [] array, Scanner s){
        return fill(array, s, 0);
    }
    private static int fill(int [] array, Scanner s, int start){
        if(start == array.length)
            return 0;
        System.out.print("Enter a number: ");
        int temp = s.nextInt();
        if(temp == -1)
            return 0;
        else{
            array[start] = temp;
            return 1 + fill(array, s, start+1);
        }
    }

    public static void print(int array[]){
        System.out.print("[");
        print(array, 0);
        System.out.print("]");
    }
    private static void print(int array[], int start){
        System.out.print(array[start]);
        if(start + 1 != array.length && array[start+1] != -1){
            System.out.print(", ");
            print(array, start+1);
        }
    }

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

```

public static int countOcc(int array[], int x){
    return countOcc(array, x, 0);
}
private static int countOcc(int array[], int x, int start){
    if(start == array.length || array[start] == -1)
        return 0;
    if(array[start] == x)
        return 1 + countOcc(array, x, start+1);
    return countOcc(array, x, start+1);
}

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

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

public static int[] mergeTwo(int a[], int b[]){
    if(!isSorted(a) || !isSorted(b))
        return null;
    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;
        else{
            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 array[], int x){
    if(!isSorted(array))
        return -1;
    return binarySearch(array, x, 0, array.length-1);
}

private static int binarySearch(int array[], int x,
                                int start, int end){
    if(start > end)
        return -1;
    int mid = (start+end)/2;
    if(array[mid] == x)
        return mid;
    else if(array[mid] < x)
        return binarySearch(array, x, mid+1, end);
    else
        return binarySearch(array, 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:
    for(int i = 0; i < array.length; i++)
        array[i] = -1;
    fill(array, input);
    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 occurred " + 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: ");
    for(int i = 0; i < b.length; i++){
        b[i] = input.nextInt();
    }
    int c[] = mergeTwo(array,b);
    if(c == null)
        System.out.println("The two arrays are 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!");
}
}
```

Sample Run

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 1

Enter a number: 1

Enter a number: 2

Enter a number: 3

Enter a number: 4

Enter a number: 5

Enter a number: 6

Enter a number: 7

Enter a number: 8

Enter a number: 9

Enter a number: 10

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 2

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 3

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 3

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 4

Enter x: 2

The number 2 has occurred 1 times in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 5

The array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] is not Palindrome

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 6

Enter the new array size: 4

Enter the array numbers: 0 3 6 12

Merged: [0, 1, 2, 3, 3, 4, 5, 6, 6, 7, 8, 9, 10, 12]

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.
- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 7

Enter number: 5

The number 5 is at index 4

- 1) Enter a new array.
- 2) Print the current array.
- 3) Reverse current array.
- 4) Count occurrences.

- 5) Check if array is palindrome
- 6) Merge with another sorted array.
- 7) Search in the sorted array.
- 8) Quit.

Enter a choice: 1

Enter a number: 1

Enter a number: 2

Enter a number: 3

Enter a number: 4

Enter a number: 5

Enter a number: 5

Enter a number: 4

Enter a number: 3

Enter a number: 2

Enter a number: 1

1) Enter a new array.

2) Print the current array.

3) Reverse current array.

4) Count occurrences.

5) Check if array is palindrome

6) Merge with another sorted array.

7) Search in the sorted array.

8) Quit.

Enter a choice: 5

The array [1, 2, 3, 4, 5, 5, 4, 3, 2, 1] is Palindrome

1) Enter a new array.

2) Print the current array.

3) Reverse current array.

4) Count occurrences.

5) Check if array is palindrome

6) Merge with another sorted array.

7) Search in the sorted array.

8) Quit.

Enter a choice: 8

Bye!