SBA-8 Abdul Javid, 210932

1. program to take input of two integer arrays from the user and to find the sum of both the arrays.
Sort the elements of the resultant array in ascending order using selection sort.

```java
import java.util.*;
public class ArrayOne {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array size");
        int size = sc.nextInt();
        int[] arr1 = new int[size];
        int[] arr2 = new int[size];
        int[] sum = new int[size];
        System.out.println("Enter the elements for array 1: ");
        for(int i=0;i<size;i++)
        {
            arr1[i] = sc.nextInt();
        }
        System.out.println("Enter the elements for array 2: ");
        for(int i=0;i<size;i++)
        {
            arr2[i] = sc.nextInt();
        }
        for(int i=0;i<size;i++)
        {
            sum[i]= arr1[i]+arr2[i];
        }
        System.out.println("The resultant array elements before sorting : ");
        System.out.print(Arrays.toString(sum));
        for(int i=0;i<size-1;i++)
        {
            int min=i;
            for(int j=i+1;j<size;j++)
            {

                if(sum[j]<sum[min])
                {
                    min=j;
                }
            }
            int temp = sum[i];
            sum[i] = sum[min];
            sum[min] = temp;
        }
        System.out.println();
        System.out.println("Using selection sorting resultant array is : ");
        System.out.print(Arrays.toString(sum));
        sc.close();
    }

}
```

Output:

```
Enter the array size
6
Enter the elements for array 1:
8
1
6
4
9
0
Enter the elements for array 2:
4
7
3
2
0
1
The resultant array elements before sorting :
[12, 8, 9, 6, 9, 1]
Using selection sorting resultant array is :
[1, 6, 8, 9, 9, 12]
```

2.program to take input of Two arrays and store the similar elements into the resultant array.
 sort the resultant array in ascending order using bubble sort.
NOTE: there must at least be 6 similar elements.
similar elements= the elements occurring in both the arrays.

```java
import java.util.Scanner;

public class ArrayTwoSort {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array size");
        int size = sc.nextInt();
        int[] arr1 = new int[size];
        int[] arr2 = new int[size];
        int[] arr3 = new int[size];
        int count = 0;
        System.out.println("Enter the elements for array 1: ");
        for(int i=0;i<size;i++)
        {
            arr1[i] = sc.nextInt();
        }
        System.out.println("Enter the elements for array 2: ");
        for(int i=0;i<size;i++)
        {
            arr2[i] = sc.nextInt();
        }

        for(int x=0;x<size;x++)
        {
            for(int y=0;y<size;y++)
            {
                if(arr1[x]==arr2[y])
                {
                    arr3[count]=arr2[y];
                    count++;
                }
            }
        }
        int len = count;
        System.out.println();
        System.out.println("The resultant array before sorting  : ");
        for(int k=0;k<len;k++)
        {
            System.out.print(arr3[k]+" ");
        }
        for(int i=0;i<len-1;i++)
        {
            for(int j=0;j<len-i-1;j++)
            {
                if(arr3[j] > arr3[j+1])
                {
                    int temp = arr3[j];
                    arr3[j] = arr3[j+1];
                    arr3[j+1] = temp;
                }
            }
        }
        System.out.println();
        System.out.println("After bubble sorting: ");
        for(int k=0;k<len;k++)
        {
            System.out.print(arr3[k]+" ");
        }
        sc.close();
    }
}
```

Output:

```
Enter the array size
10
Enter the elements for array 1:
7
1
0
15
20
6
9
3
10
5
Enter the elements for array 2:
14
20
3
9
15
27
18
0
1
8
The resultant array before sorting  :
1 0 15 20 9 3
After bubble sorting:
0 1 3 9 15 20
```

3.program to take input two arrays and store the dissimilar elements into a resultant array.
sort the resultant array in a descending order using bubble sort.
dissimilar elements= the elements not occurring in both the arrays. (unique elements)

```java
import java.util.Arrays;
import java.util.Scanner;
import java.util.TreeSet;
public class ArrayThreeSort {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array size");
        int size = sc.nextInt();
        int[] arr1 = new int[size];
        int[] arr2 = new int[size];
        System.out.println("Enter the elements for array 1: ");
        for(int i=0;i<size;i++)
        {
            arr1[i] = sc.nextInt();
        }
        System.out.println("Enter the elements for array 2: ");
        for(int i=0;i<size;i++)
        {
            arr2[i] = sc.nextInt();
        }
        TreeSet<Integer> set = new TreeSet<>();
        for(int x:arr1)
        {
            set.add(x);
        }
        for(int y:arr2)
        {
            set.add(y);
        }
        System.out.println();
        System.out.println("The resultant array before sorting  : ");
        int len = set.size();
        Integer[] res = new Integer[len];
        res = set.toArray(res);
        System.out.println(Arrays.toString(res));
        for(int i=0;i<len-1;i++)
        {
            for(int j=0;j<len-i-1;j++)
            {
                if(res[j] < res[j+1])
                {
                    int temp = res[j];
                    res[j] = res[j+1];
                    res[j+1] = temp;
                }
            }
        }
        System.out.println("After sorting: ");
        System.out.println(Arrays.toString(res));
        sc.close();
    }
}
```

Output:

```
Enter the array size
4
Enter the elements for array 1:
1
2
3
4
Enter the elements for array 2:
5
6
2
7
The resultant array before sorting  :
[1, 2, 3, 4, 5, 6, 7]
After sorting:
[7, 6, 5, 4, 3, 2, 1]
```

4. Implement Array List and add, remove, elements in the Array List and perform sorting of the elements using the iterator.

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
public class ArrayListIterator {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(15);
        list.add(30);
        list.add(20);
        list.add(5);
        // Removing the element at 3rd position
        list.remove(3);
        // Displaying the list
        System.out.println("The list is: \n"+ list);
        // Create an iterator for the list
        // using iterator() method
        Iterator<Integer> iter = list.iterator();
        // Displaying the values
        // after iterating through the list
        System.out.println("\nThe iterator values" + " of list are: ");
        while (iter.hasNext()) {
            System.out.print(iter.next() + " ");
        }
        Collections.sort(list);
        System.out.println();
        System.out.println("After sorting");
        System.out.println(list);
    }
}
```

Output:

```
The list is:
[10, 15, 30, 5]

The iterator values of list are:
10 15 30 5
After sorting
[5, 10, 15, 30]
```

5. Implement LinkedList and add, remove, elements in the LinkedList and perform sorting of the elements using the iterator.

```java
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;

public class LinkedListIterator {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        LinkedList<String> list = new LinkedList<>();
        list.add("hit");
        list.add("bit");
        list.add("wit");
        list.add("pit");
        list.add("sit");
        System.out.println("LinkedList: "+list);
        list.remove(2);
        System.out.println("LinkedList After deletion: "+list);

        Iterator<String> itr = list.iterator();
        System.out.println("The iterator values of list are: ");
        while(itr.hasNext())
        {
            System.out.print(itr.next()+" ");
        }
        System.out.println();
        Collections.sort(list);
        System.out.println("After Sorting LinkedList: ");
        System.out.println(list);
    }
}
```

Output:

```
LinkedList: [hit, bit, wit, pit, sit]
LinkedList After deletion: [hit, bit, pit, sit]
The iterator values of list are:
hit bit pit sit
After Sorting LinkedList:
[bit, hit, pit, sit]
```