

Divyanshu Sharma
199303003
Sec A
CCE

OOPS ASSIGNMENT 5

Q16

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.stream.Collectors;
import java.util.Scanner;

class Main {
    ArrayList<Integer> arrList;

    Main(ArrayList<Integer> arrList) {
        this.arrList = arrList;
    }

    List<Integer> distinct() {
        return arrList.stream().distinct().collect(Collectors.toList());
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> arrayList = new ArrayList<Integer>();
        int temp;
        while (sc.hasNext()) {
            temp = sc.nextInt();
            arrayList.add(temp);
        }

        Main m = new Main(arrayList);
        List<Integer> li = m.distinct();

        String str = li.stream()
            .map(i -> i.toString())
            .collect(Collectors.joining(" "));
        System.out.println(str);
    }
}
```

[Back to Classroom](#)

Submit Before: 1/29/2021, 9:28:00 AM

Q16
Write a program to remove all duplicate elements from ArrayList and keeping unique elements.
input: 1 2 3 4 2 22 11 2 1
output: 1 2 3 4 22 11

Submitted version (2 passed of 2) -

RunDebugStopTest

Language: Java

Test Result

Pass
2 test(s) passed out of 2 test(s)

OK

```
1 import java.util.ArrayList;
2 import java.util.Comparator;
3 import java.util.List;
4 import java.util.Scanner;
5 import java.util.stream.Collectors;
6
7 class Main {
8     ArrayList<Integer> arrList;
9
10    Main(ArrayList<Integer> arrList) {
11        this.arrList = arrList;
12    }
13
14    List<Integer> distinct() {
15        return arrList.stream().distinct().collect(Collectors.toList());
16    }
17
18    public static void main(String args[]) {
19        Scanner sc = new Scanner(System.in);
20        ArrayList<Integer> arrayList = new ArrayList<Integer>();
21        int temp;
22        while (sc.hasNext()) {
23            temp = sc.nextInt();
24            arrayList.add(temp);
25        }
26    }
27 }
```

Q17

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.stream.Collectors;
import java.util.Scanner;

class Main {
    ArrayList<Integer> arrList;

    Main(ArrayList<Integer> arrList) {
        this.arrList = arrList;
    }

    void sort() {
        arrList.sort(Comparator.naturalOrder());
    }

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> arrayList = new ArrayList<Integer>();
        int temp;
        while (sc.hasNext()) {
            temp = sc.nextInt();
            arrayList.add(temp);
        }
        Main m = new Main(arrayList);
        m.sort();

        String str = m.arrList.stream()
            .map(i -> i.toString())
```

```

        .collect(Collectors.joining(" "));
    System.out.println(str);
}
}

```

The screenshot shows a Java IDE interface. On the left, a panel for 'Q17' contains the text: 'Write a program to sort a given array list in reverse order', 'input: 1 44 1 22 3 4 66', and 'Output: 1 1 3 4 22 44 66'. The main editor displays a Java program for sorting an ArrayList in reverse order. A 'Test Result' dialog box is open, showing 'Pass' and '1 test(s) passed out of 1 test(s)'. The code in the editor is as follows:

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.List;
4 import java.util.Scanner;
5 import java.util.Scanner;
6
7 class Main {
8     ArrayList<Integer> arrList;
9
10    Main(ArrayList<Integer> arrList) {
11        this.arrList = arrList;
12    }
13
14    void sort() {
15        arrList.sort(Collections.reverseOrder());
16    }
17
18    public static void main(String[] args) {

```

Q18

```

import java.util.*;

class Main {
    public static void main(String args[]) {
        LinkedList<Integer> ll = new LinkedList<Integer>();

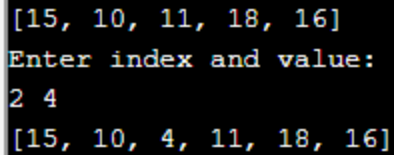
        ll.add(15);
        ll.add(10);
        ll.add(11);
        ll.add(18);
        ll.add(16);

        Scanner sc = new Scanner(System.in);

        System.out.println(ll);
        System.out.println("Enter index and value:");
        int i = sc.nextInt();
        int x = sc.nextInt();
        ll.add(i, x);

        System.out.println(ll);
    }
}

```

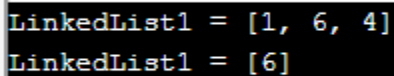
A terminal window with a dark background. It shows an array [15, 10, 11, 18, 16], a prompt "Enter index and value:", the input "2 4", and the resulting array [15, 10, 4, 11, 18, 16].

```
[15, 10, 11, 18, 16]
Enter index and value:
2 4
[15, 10, 4, 11, 18, 16]
```

Q19

```
import java.util.LinkedList;

class Main {
    public static void main(String args[]) {
        LinkedList<Integer> l11 = new LinkedList<Integer>();
        l11.add(1);
        l11.add(6);
        l11.add(4);
        System.out.println("LinkedList1 = " + l11);
        LinkedList<Integer> l12 = new LinkedList<Integer>(l11);
        l12.removeFirst();
        l12.removeLast();
        System.out.println("LinkedList1 = " + l12);
    }
}
```

A terminal window with a dark background. It shows the output of the previous code: "LinkedList1 = [1, 6, 4]" followed by "LinkedList1 = [6]".

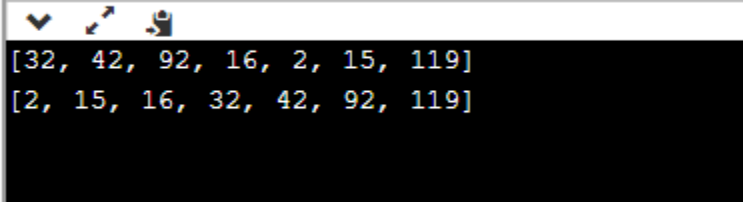
```
LinkedList1 = [1, 6, 4]
LinkedList1 = [6]
```

Q20

```
import java.util.Arrays;

class Main {
    public static void main(String args[]) {
        int arr[] = {32, 42, 92, 16, 2, 15, 119};
        System.out.println(Arrays.toString(arr));
        bubbleSort(arr);
        System.out.println(Arrays.toString(arr));
    }
}
```

```
static void bubbleSort(int arr[])
{
    int n = arr.length;
    for (int i = 0; i < n-1; i++)
        for (int j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
            {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
}
```



A terminal window with a dark background and light-colored text. At the top, there are three small icons: a downward arrow, a magnifying glass, and a document icon. Below the icons, the terminal displays two lines of text representing an array. The first line is "[32, 42, 92, 16, 2, 15, 119]" and the second line is "[2, 15, 16, 32, 42, 92, 119]".

```
[32, 42, 92, 16, 2, 15, 119]
[2, 15, 16, 32, 42, 92, 119]
```