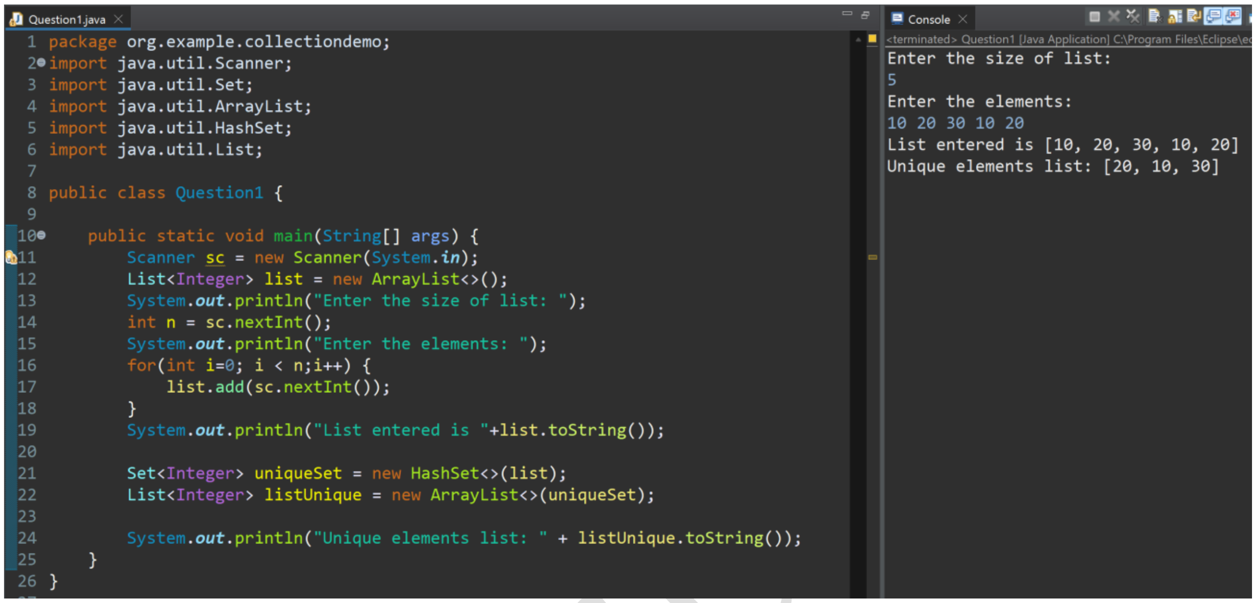
PG-DAC AUGUST 24 BATCH

1)Write a Java program that takes a list of integers as input and returns a list of duplicate integers.



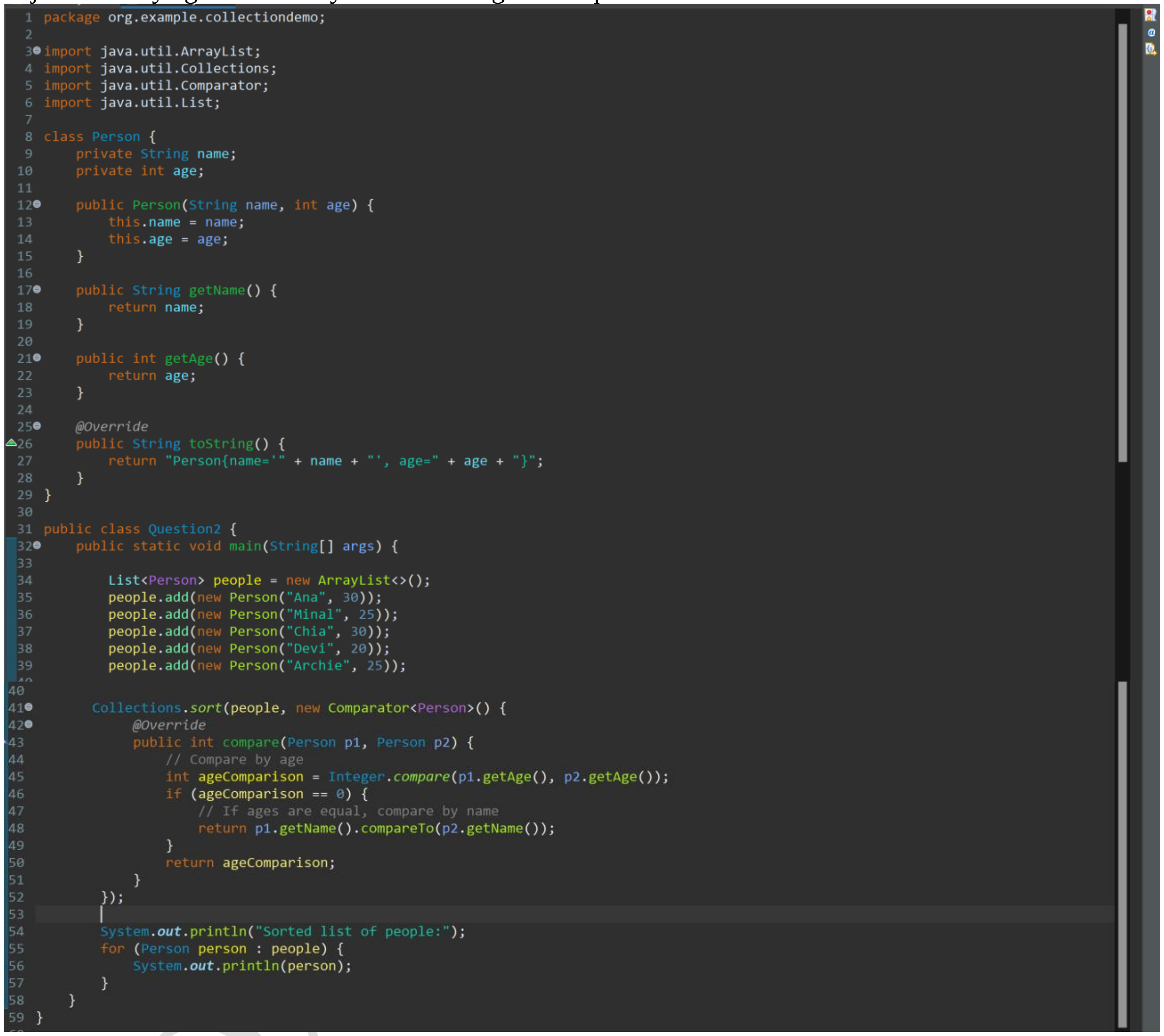
```
package org.example.collectiondemo;
import java.util.Scanner;
import java.util.Set;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;

public class Question1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        List<Integer> list = new ArrayList<>();
        System.out.println("Enter the size of list: ");
        int n = sc.nextInt();
        System.out.println("Enter the elements: ");
        for(int i=0; i < n;i++) {
            list.add(sc.nextInt());
        }
        System.out.println("List entered is "+list.toString());

        Set<Integer> uniqueSet = new HashSet<>(list);
        List<Integer> listUnique = new ArrayList<>(uniqueSet);

        System.out.println("Unique elements list: " + listUnique.toString());
    }
}
```

```
<terminated> Question1 [Java Application] C:\Program Files\Eclipse\e
Enter the size of list:
5
Enter the elements:
10 20 30 10 20
List entered is [10, 20, 30, 10, 20]
Unique elements list: [20, 10, 30]
```

Code:

```
package org.example.collectiondemo;
import java.util.Scanner;
import java.util.Set;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
public class Question1 {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		List<Integer> list = new ArrayList<>();
		System.out.println("Enter the size of list: ");
		int n = sc.nextInt();
		System.out.println("Enter the elements: ");
		for(int i=0; i < n;i++) {
			list.add(sc.nextInt());
		}
		System.out.println("List entered is "+list.toString());

		Set<Integer> uniqueSet = new HashSet<>(list);
		List<Integer> listUnique = new ArrayList<>(uniqueSet);

		System.out.println("Unique elements list: " + listUnique.toString());
	}
}
```

2)Create a Person class with attributes name and age. Write a Java program that sorts a list of Person objects first by age and then by name if the ages are equal.



```java
package org.example.collectiondemo;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    @Override
    public String toString() {
        return "Person{name='" + name + "', age=" + age + "}";
    }
}

public class Question2 {
    public static void main(String[] args) {

        List<Person> people = new ArrayList<>();
        people.add(new Person("Ana", 30));
        people.add(new Person("Minal", 25));
        people.add(new Person("Chia", 30));
        people.add(new Person("Devi", 20));
        people.add(new Person("Archie", 25));

      Collections.sort(people, new Comparator<Person>() {
            @Override
            public int compare(Person p1, Person p2) {
                // Compare by age
                int ageComparison = Integer.compare(p1.getAge(), p2.getAge());
                if (ageComparison == 0) {
                    // If ages are equal, compare by name
                    return p1.getName().compareTo(p2.getName());
                }
                return ageComparison;
            }
        });

        System.out.println("Sorted list of people:");
        for (Person person : people) {
            System.out.println(person);
        }
    }
}
```

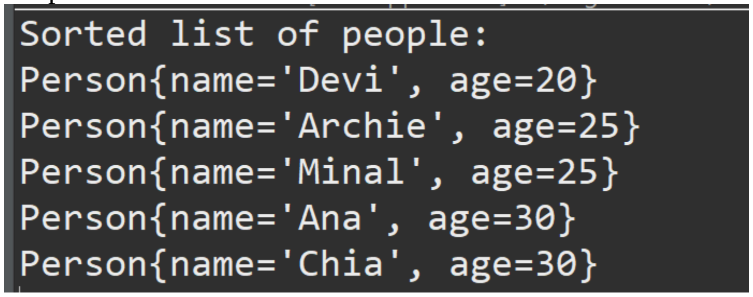
Code:
package org.example.collectiondemo;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

```java
class Person {
   private String name;
   private int age;

   public Person(String name, int age) {
      this.name = name;
```

```
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    @Override
    public String toString() {
        return "Person{name='" + name + "', age=" + age + "}";
    }
}

public class Question2 {
    public static void main(String[] args) {

        List<Person> people = new ArrayList<>();
        people.add(new Person("Ana", 30));
        people.add(new Person("Minal", 25));
        people.add(new Person("Chia", 30));
        people.add(new Person("Devi", 20));
        people.add(new Person("Archie", 25));

       Collections.sort(people, new Comparator<Person>() {
            @Override
            public int compare(Person p1, Person p2) {
                // Compare by age
                int ageComparison = Integer.compare(p1.getAge(), p2.getAge());
                if (ageComparison == 0) {
                    // If ages are equal, compare by name
                    return p1.getName().compareTo(p2.getName());
                }
                return ageComparison;
            }
        });

        System.out.println("Sorted list of people:");
        for (Person person : people) {
            System.out.println(person);
        }
    }
}
```

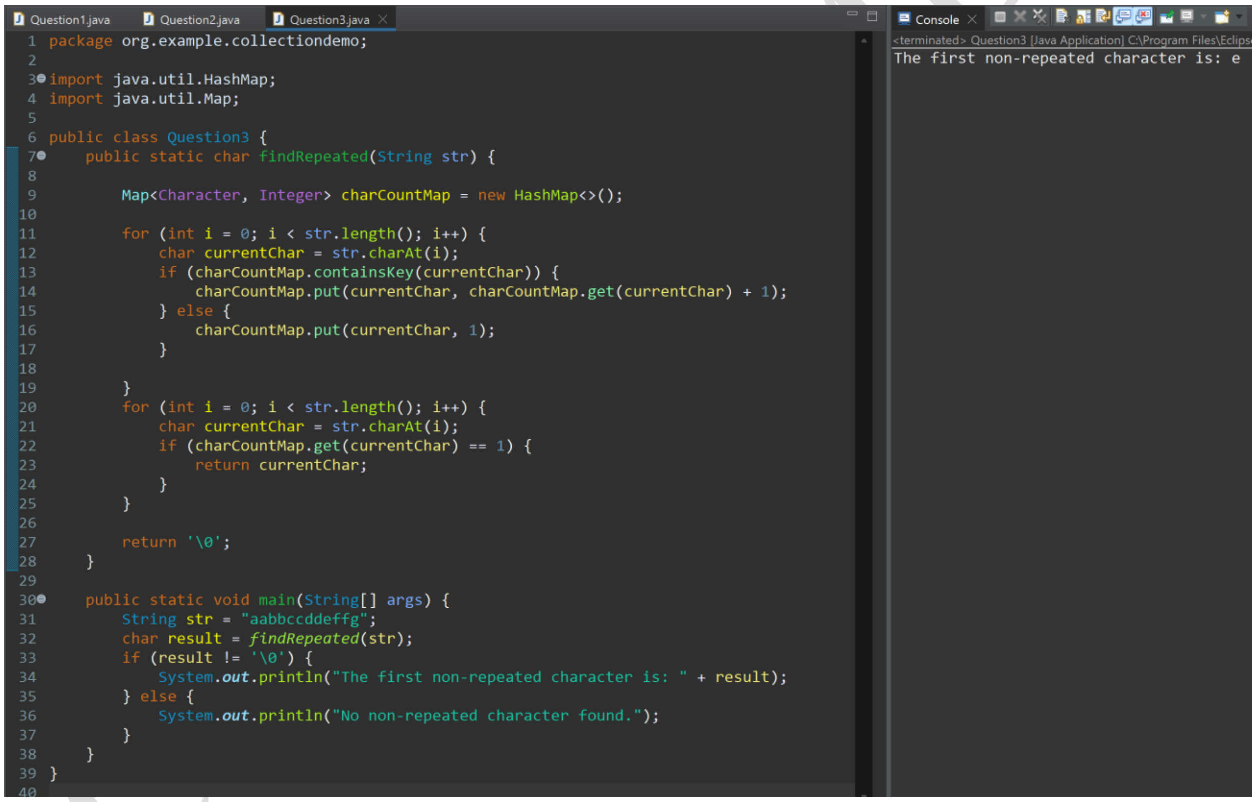
Output:



```
Sorted list of people:
Person{name='Devi', age=20}
Person{name='Archie', age=25}
Person{name='Minal', age=25}
Person{name='Ana', age=30}
Person{name='Chia', age=30}
```

3)Write a Java program to find the first non-repeated character in a string using a HashMap.

String input = "aabbccddeffg";
Expected output = 'e';


Question1.java
Question2.java
Question3.java
Console
<terminated> Question3 [Java Application] C:\Program Files\Eclips

```java
package org.example.collectiondemo;

import java.util.HashMap;
import java.util.Map;

public class Question3 {
    public static char findRepeated(String str) {

        Map<Character, Integer> charCountMap = new HashMap<>();

        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            if (charCountMap.containsKey(currentChar)) {
                charCountMap.put(currentChar, charCountMap.get(currentChar) + 1);
            } else {
                charCountMap.put(currentChar, 1);
            }

        }
        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            if (charCountMap.get(currentChar) == 1) {
                return currentChar;
            }
        }

        return '\0';
    }

    public static void main(String[] args) {
        String str = "aabbccddeffg";
        char result = findRepeated(str);
        if (result != '\0') {
            System.out.println("The first non-repeated character is: " + result);
        } else {
            System.out.println("No non-repeated character found.");
        }
    }
}
```

```
The first non-repeated character is: e
```

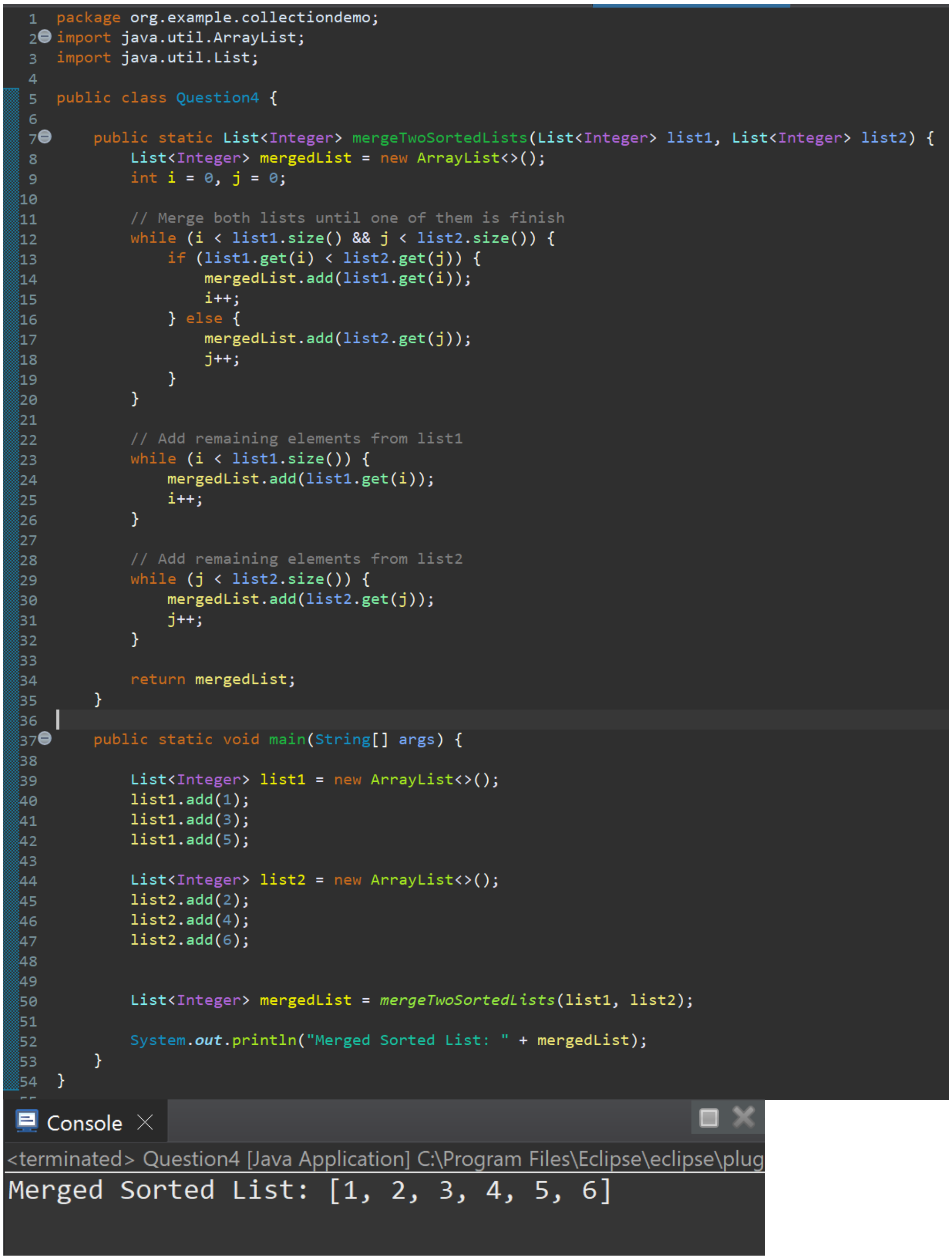
Code: package org.example.collectiondemo;

```java
import java.util.HashMap;
import java.util.Map;

public class Question3 {
    public static char findRepeated(String str) {

        Map<Character, Integer> charCountMap = new HashMap<>();
        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            if (charCountMap.containsKey(currentChar)) {
                charCountMap.put(currentChar, charCountMap.get(currentChar) + 1);
            } else {
                charCountMap.put(currentChar, 1);
            }

        }
        for (int i = 0; i < str.length(); i++) {
            char currentChar = str.charAt(i);
            if (charCountMap.get(currentChar) == 1) {
                return currentChar;
            }
        }

        return '\0';
    }

    public static void main(String[] args) {
        String str = "aabbccddeffg";
        char result = findRepeated(str);
        if (result != '\0') {
            System.out.println("The first non-repeated character is: " + result);
        } else {
            System.out.println("No non-repeated character found.");
        }
    }
}
```

4) Write a Java program that merges two sorted lists of integers into a single sorted list.

Code:

```
package org.example.collectiondemo;
import java.util.ArrayList;
import java.util.List;

public class Question4 {

    public static List<Integer> mergeTwoSortedLists(List<Integer> list1, List<Integer> list2) {
        List<Integer> mergedList = new ArrayList<>();
        int i = 0, j = 0;

        // Merge both lists until one of them is finish
        while (i < list1.size() && j < list2.size()) {
            if (list1.get(i) < list2.get(j)) {
                mergedList.add(list1.get(i));
                i++;
            } else {
                mergedList.add(list2.get(j));
                j++;
            }
        }

        // Add remaining elements from list1
        while (i < list1.size()) {
            mergedList.add(list1.get(i));
            i++;
        }

        // Add remaining elements from list2
        while (j < list2.size()) {
            mergedList.add(list2.get(j));
            j++;
        }

        return mergedList;
    }

    public static void main(String[] args) {

        List<Integer> list1 = new ArrayList<>();
        list1.add(1);
        list1.add(3);
        list1.add(5);

        List<Integer> list2 = new ArrayList<>();
        list2.add(2);
        list2.add(4);
        list2.add(6);
        List<Integer> mergedList = mergeTwoSortedLists(list1, list2);

        System.out.println("Merged Sorted List: " + mergedList);
    }
}
```



```java
package org.example.collectiondemo;
import java.util.ArrayList;
import java.util.List;

public class Question4 {

    public static List<Integer> mergeTwoSortedLists(List<Integer> list1, List<Integer> list2) {
        List<Integer> mergedList = new ArrayList<>();
        int i = 0, j = 0;

        // Merge both lists until one of them is finish
        while (i < list1.size() && j < list2.size()) {
            if (list1.get(i) < list2.get(j)) {
                mergedList.add(list1.get(i));
                i++;
            } else {
                mergedList.add(list2.get(j));
                j++;
            }
        }

        // Add remaining elements from list1
        while (i < list1.size()) {
            mergedList.add(list1.get(i));
            i++;
        }

        // Add remaining elements from list2
        while (j < list2.size()) {
            mergedList.add(list2.get(j));
            j++;
        }

        return mergedList;
    }

    public static void main(String[] args) {

        List<Integer> list1 = new ArrayList<>();
        list1.add(1);
        list1.add(3);
        list1.add(5);

        List<Integer> list2 = new ArrayList<>();
        list2.add(2);
        list2.add(4);
        list2.add(6);


        List<Integer> mergedList = mergeTwoSortedLists(list1, list2);

        System.out.println("Merged Sorted List: " + mergedList);
    }
}
```

Console

```
<terminated> Question4 [Java Application] C:\Program Files\Eclipse\eclipse\plug
Merged Sorted List: [1, 2, 3, 4, 5, 6]
```