**Assignment On Collection**

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

**package** IntegerDuplicate;

**import** java.util.\*;

**public** **class** DuplicateFinder {

**public** **static** List<Integer> findDuplicates(List<Integer> numbers) {

Set<Integer> uniqueNumbers = **new** HashSet<>();

List<Integer> duplicates = **new** ArrayList<>();

**for** (Integer num : numbers) {

**if** (!uniqueNumbers.add(num)) {

**if** (!duplicates.contains(num)) {

duplicates.add(num);

}

}

}

**return** duplicates;

}

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

List<Integer> numbers = **new** ArrayList<>();

System.***out***.println("Enter numbers (type 'done' to finish):");

**while** (scanner.hasNextInt()) {

numbers.add(scanner.nextInt());

}

List<Integer> duplicates = *findDuplicates*(numbers);

System.***out***.println("Duplicate numbers: " + duplicates);

}

}

Output:

Enter numbers (type 'done' to finish):

1 2 3 4 5 3 2 7 8 5

done

Duplicate numbers: [3, 2, 5]

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.

**package** IntegerDuplicate;

**import** java.util.\*;

**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** PersonSorter {

**public** **static** **void** main(String[] args) {

List<Person> people = **new** ArrayList<>();

people.add(**new** Person("Alice", 30));

people.add(**new** Person("Bob", 25));

people.add(**new** Person("Charlie", 30));

people.add(**new** Person("David", 20));

people.add(**new** Person("Eve", 25));

Collections.*sort*(people, **new** Comparator<Person>() {

@Override

**public** **int** compare(Person p1, Person p2) {

**int** ageCompare = Integer.*compare*(p1.getAge(), p2.getAge());

**if** (ageCompare == 0) {

**return** p1.getName().compareTo(p2.getName());

} **else** {

**return** ageCompare;

}

}

});

System.***out***.println("Sorted list of people:");

**for** (Person person : people) {

System.***out***.println(person);

}

}

}

Output:

Sorted list of people:

Person{name='David', age=20}

Person{name='Bob', age=25}

Person{name='Eve', age=25}

Person{name='Alice', age=30}

Person{name='Charlie', 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';

Program code:

**package** IntegerDuplicate;

**import** java.util.HashMap;

**public** **class** FirstNonRepeatedCharacter {

**public** **static** Character findFirstNonRepeatedChar(String input) {

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

**for** (**char** c : input.toCharArray()) {

charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);

}

**for** (**char** c : input.toCharArray()) {

**if** (charCountMap.get(c) == 1) {

**return** c;

}

}

**return** **null**;

}

**public** **static** **void** main(String[] args) {

String input = "aabbccddeffg";

Character result = *findFirstNonRepeatedChar*(input);

**if** (result != **null**) {

System.***out***.println("The first non-repeated character is: " + result);

} **else** {

System.***out***.println("No non-repeated character found.");

}

}

}

Output:

The first non-repeated character is: e

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

Program code:

**package** IntegerDuplicate;

**import** java.util.\*;

**public** **class** MergeSortedLists {

**public** **static** List<Integer> mergeSortedLists(List<Integer> list1, List<Integer> list2) {

List<Integer> mergedList = **new** ArrayList<>();

**int** i = 0, j = 0;

**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++;

}

}

**while** (i < list1.size()) {

mergedList.add(list1.get(i));

i++;

}

**while** (j < list2.size()) {

mergedList.add(list2.get(j));

j++;

}

**return** mergedList;

}

**public** **static** **void** main(String[] args) {

List<Integer> list1 = Arrays.*asList*(1, 3, 5, 7);

List<Integer> list2 = Arrays.*asList*(2, 4, 6, 8);

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

System.***out***.println("Merged sorted list: " + mergedList);

}

}

Output:

Merged sorted list: [1, 2, 3, 4, 5, 6, 7, 8]