**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.

**Code:**

package collection;

import java.util.\*;

public class list {

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

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

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

for (Integer number : numbers) {

if (!uniqueNumbers.add(number)) {

duplicates.add(number);

}

}

return duplicates;

}

public static void main(String[] args) {

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

System.***out***.print("Enter the number of elements: ");

int n = scanner.nextInt();

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

System.***out***.println("Enter the integers:");

for (int i = 0; i < n; i++) {

numbers.add(scanner.nextInt());

}

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

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

}

}

**O/p:**

Enter the number of elements: 6

Enter the integers:

1

1

2

4

5

7

Duplicate integers: [1]

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.

**Code:**

package collection;

import java.util.\*;

class Main {

private String name;

private int age;

// Constructor

public Main(String name, int age) {

this.name = name;

this.age = age;

}

// Getters

public String getName() {

return name;

}

public int getAge() {

return age;

}

// toString method to display Person details

*@Override*

public String toString() {

return "Person{name='" + name + "', age=" + age + "}";

}

}

public class person {

public static void main(String[] args) {

// Creating a list of Person objects

List<Main> people = new ArrayList<>();

people.add(new Main("Yash", 25));

people.add(new Main("Vaibhav", 30));

people.add(new Main("Saurabh", 25));

people.add(new Main("Samarth", 20));

// Sorting logic: First by age, then by name if ages are equal

people.sort(Comparator.*comparingInt*(Main::getAge)

.thenComparing(Main::getName));

// Displaying the sorted list

System.***out***.println("Sorted List of People:");

for (Main person : people) {

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

}

}

}

**O/p:**

Sorted List of People:

Person{name=’Samarth', age=20}

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

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

Person{name=’Vaibhav ', 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';

**Code:**

package collection;

import java.util.HashMap;

import java.util.Map;

public class FirstNonRepeatedCharacter {

public static void main(String[] args) {

String input = "aabbccddeffg";

char result = *findFirstNonRepeatedCharacter*(input);

if (result != '\0') {

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

} else {

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

}

}

private static char findFirstNonRepeatedCharacter(String str) {

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

for (char ch : str.toCharArray()) {

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

}

for (char ch : str.toCharArray()) {

if (charCountMap.get(ch) == 1) {

return ch;

}

}

return '\0';

}

}

**O/p:**

The first non-repeated character is: e

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

**Code:**

package collection;

import java.util.\*;

public class MergeSortedLists {

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);

}

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;

}

}

**O/p:**

Merged Sorted List: [1, 2, 3, 4, 5, 6, 7, 8]