**1256 – HASH TABLE**

1. import java.io.BufferedReader; // import BufferedReader class

2. import java.io.IOException; // import IOException class

3. import java.io.InputStreamReader; // import InputStreamReader class

4. import java.io.PrintWriter; // import PrintWriter class

5. import java.util.Arrays; // import Arrays class

6.

7. public class HashTable { // HashTable class declaration

8.

9.     static BufferedReader inputReader = new BufferedReader(new InputStreamReader(System.in)); // instantiate

10.                                                                                                // BufferedReader

11.                                                                                                // object

12.     static PrintWriter outputWriter = new PrintWriter(System.out); // instantiate PrintWriter object

13.

14.     public static void main(String[] args) throws IOException { // main method declaration

15.         int testCases = readInt(); // read number of test cases

16.

17.         while (testCases-- > 0) { // loop through test cases

18.             int[] parameters = readIntArray(); // read parameters

19.             int size = parameters[0]; // read size

20.             String[] hashTable = new String[size]; // instantiate hash table

21.             Arrays.fill(hashTable, ""); // fill hash table with empty strings

22.

23.             String[] elements = readLine().split("\\s"); // read elements

24.             for (String element : elements) { // loop through elements

25.                 int index = toInt(element) % size; // calculate index

26.                 hashTable[index] += " -> " + element; // add element to hash table

27.             } // end of loop through elements

28.

29.             for (int i = 0; i < size; i++) { // loop through hash table

30.                 outputWriter.println(i + hashTable[i] + " -> \\"); // print hash table

31.             } // end of loop through hash table

32.

33.             if (testCases != 0) { // if it is not the last test case

34.                 outputWriter.println(); // print blank line

35.             } // end of if

36.         } // end of loop through test cases

37.

38.         outputWriter.close();

39.     } // end of main method

40.

41.     private static String readLine() throws IOException {

42.         return inputReader.readLine();

43.     }

44.

45.     private static int readInt() throws IOException {

46.         return Integer.parseInt(inputReader.readLine());

47.     }

48.

49.     private static int[] readIntArray() throws IOException {

50.         String[] line = inputReader.readLine().split("\\s");

51.         int length = line.length;

52.         int[] array = new int[length];

53.         for (int i = 0; i < length; i++) {

54.             array[i] = Integer.parseInt(line[i]);

55.         }

56.         return array;

57.     }

58.

59.     private static int toInt(String s) {

60.         return Integer.parseInt(s);

61.     }

62. } // end of HashTable class

63.

**EXPLICAÇÃO DO ALGORITMO**