Java Lab CAT1

Name: Sankalp Mukim

Registration Number: 20BDS0128

Slot: L21+L22

Table of Contents

[Question: 1](#_Toc83893119)

[Code: 2](#_Toc83893120)

[Class NNumber: 2](#_Toc83893121)

[Class MainClass: 5](#_Toc83893122)

[Output: 7](#_Toc83893123)

# Question:

Define a class NNumber with an instance variable, n (make sure that the number contains atleast 3 digits), a String array to store the word equivalents of the individual digits and another variable, shuffle. Define the constructor that assigns the user input to the instance variable and stores word equivalents of the digits of the number. Define a method sort() that sorts the words in the String array and accordingly shuffles the digits in the number and stores them in shuffle. Define another method that counts the number of digits that are retained in the same positions even after sorting. Instantiate this class in the main class and invoke the appropriate methods for ‘m’ objects.

Sample Input:

3 (value of m)

79058

2003

463

The corresponding String array of each object would be as follows

{'seven', 'nine', 'zero', 'five', 'eight'}

{'two', 'zero', 'zero', 'three'}

{'four', 'six', 'three'}

Output:

Now after sorting, the string arrays will be

{ 'eight', 'five', 'nine', 'seven', 'zero'}

{'three', 'two', 'zero', 'zero'}

{'four', 'six', 'three'}

and so the output should be

85970

3200

463

# Code:

## Class NNumber:

public class NNumber {

public int n;

public int afterSorting;

public String[] arr;

private String[] digits = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" };

private static int numDigits(int n) {

int count = 0;

while (n > 0) {

count++;

n = n / 10;

}

return count;

}

private static int indexOf(String item, String[] arr) {

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

if (arr[i].equals(item)) {

return i;

}

}

return -1;

}

NNumber(int n) {

this.n = n;

arr = new String[numDigits(n)];

if (numDigits(n) < 3) {

System.out.println("Error!! Number of digits is less than 3");

} else {

int x = n;

int i = 0;

while (x > 0) {

arr[i] = digits[x % 10];

i++;

x /= 10;

}

// Reversing the obtained array

for (int j = 0; j < arr.length / 2; j++) {

String temp = arr[arr.length - 1 - j];

arr[arr.length - 1 - j] = arr[j];

arr[j] = temp;

}

afterSorting = 0;

}

}

public void sort() {

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

for (int j = i + 1; j < arr.length; j++) {

if (arr[i].compareTo(arr[j]) > 0) {

String temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

for (String string : arr) {

int num = indexOf(string, digits);

afterSorting \*= 10;

afterSorting += num;

}

}

public int countSamePositions() {

int count = 0;

int x = n;

int y = afterSorting;

while (x > 0) {

if (x % 10 == y % 10) {

count++;

}

x /= 10;

y /= 10;

}

return count;

}

}

## Class MainClass:

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class MainClass {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int m = sc.nextInt();

List<NNumber> list = new ArrayList<NNumber>();

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

int n = sc.nextInt();

NNumber current = new NNumber(n);

list.add(current);

}

System.out.println("The corresponding String array of each object would be as follows");

for (int i = 0; i < list.size(); i++) {

System.out.print("{");

String[] tempArr = list.get(i).arr;

for (String string : tempArr) {

System.out.print(string + ',');

}

System.out.println("}");

}

for (int i = 0; i < list.size(); i++) {

list.get(i).sort();

}

System.out.println("Now after sorting, the string arrays will be");

for (int i = 0; i < list.size(); i++) {

System.out.print("{");

String[] tempArr = list.get(i).arr;

for (String string : tempArr) {

System.out.print(string + ',');

}

System.out.println("}");

}

System.out.println("And so the output should be");

for (int i = 0; i < list.size(); i++) {

System.out.println(list.get(i).afterSorting);

}

System.out.println("The corresponding count of unchanged values:");

for (int i = 0; i < list.size(); i++) {

System.out.println(list.get(i).countSamePositions());

}

sc.close();

}

}

# Output:

Text

Description automatically generated