INSTITUE OF TECHNICAL EDUCATION AND RESEARCH
SIKSHA 'O' ANUSANDHAN UNIVERSITY
Bhubaneswar, Odisha, India

Introduction to Computer Programming

Minor-Project-1 (2021)

(This assignment tests your knowledge in looping and string)

****

Problem: Given a large number (n>0). You need to print the unique digits present in the number **n**, and the largest number possible using these unique digits. **(You are not allowed to use array.)**

Problem Description:

In the first part of the problem, you need to print the unique digits present in a given large number. Let's consider a large number **n=12134616235835**.

The unique digits present in **n** are 1, 2, 3, 4, 6, 5, and 8.

Input the large number as a Java String, then, replace each duplicate element with a special character '*'. Now, you can count the number of characters excluding '*' to get the unique digits.

N = " 1 2 1 3 4 6 1 6 2 3 5 8 3 5 "

1st iteration:	N = " 1 2 * 3 4 6 * 6 2 3 5 8 3 5 "
2nd iteration:	N = " 1 2 * 3 4 6 * 6 * 3 5 8 3 5 "
3rd iteration:	N = " 1 2 * 3 4 6 * 6 * 3 5 8 3 5 "
4th iteration:	N = " 1 2 * 3 4 6 * 6 * * 5 8 * 5 "
5th iteration:	N = " 1 2 * 3 4 6 * 6 * * 5 8 * 5 "
6th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * 5 "
7th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * 5 "
8th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * 5 "
9th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * 5 "
10th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * 5 "
11th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * * "
12th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * * "
13th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * * "
14th iteration:	N = " 1 2 * 3 4 6 * * * * 5 8 * * "

So, the unique digits are: 1, 2, 3, 4, 6, 5, and 8.

Following String and StringBuffer class methods may be helpful for you.

1. **int length():** To find length of a string.
```
Ex:   String str="ITER";
      int len=str.length();  Gives len=4.
```
2. **char charAt(int index):** To find character an index.
```
Ex:   String str="ITER";
      System.out.println(str.charAt(2));  prints E.
```
3. **int indexOf(char ch):** To find index of a character in the string.
```
Ex:   String str="ITER";
      System.out.println(str.indexOf('T')); prints 1.
```
4. **void setCharAt(int i, char ch):** This is a method in StringBuffer class, used to replace a character in a StringBuffer object. If you use this, you have to convert String to StringBuffer.
```
Ex:   String str1="MADAM";
      StringBuffer str2=new StringBuffer(str1);
      str2.setCharAt(1,'*');
      System.out.println(str2);
It prints M*DAM.
```

In the second part of the problem, you need to find the largest number possible using the unique digits found in the first part of the problem. Here, you can create a new string containing only the unique characters of **n** (**str = "1234658**"). In each iteration of the loop, find the character having the largest value and append to a blank string **(lrg = ""),** finally**,** replace it with the character '#' in **str**.

	lrg=""	**str = "1234658**"
1st iteration:	lrg = "8"	str = "123465#"
2nd iteration:	lrg = "86"	str = "1234#5#"
3rd iteration:	lrg = "865"	str = "1234###"
4th iteration:	lrg = "8654"	str = "123####"
5th iteration:	lrg = "86543"	str = "12#####"
6th iteration:	lrg = "865432"	str = "1######"
7th iteration:	lrg = "8654321"	str = "#######"

In the end, you can convert **lrg** to long value to get the largest possible number.

Marks's distribution:

1. **[1 mark]** To input a large number.
2. **[4 marks]** To print unique digits.
3. **[4 marks]** To generate the largest number string (lrg).
4. **[1 marks]** To convert the string (lrg) into a number.

Test Cases:

Sample test case:

Enter a large number: **12134616235835**
The unique digits present in 12134616235835 are 1, 2, 3, 4, 6, 5, 8.
The largest number possible out of these unique digits is 8654321.

You need to check the following test cases, and provide the results in *programDescription.docx* file.

Test case 1: Enter a large number: **12134616235835**

Test case 2: Enter a large number: **11131116111811**

Test case 3: Enter a large number: **7**

Test case 4: Enter a large number: **11111111111**

Test case 5: Enter a large number: **1253478690**

Test case 6: Enter a large number: **000000000**

Test case 7: Enter a large number: **1222222222**

Test case 8: Enter a large number: **33333333335**

Note: Create ***uniqueDigitsOfLargeNumber.java*** (Program) and ***programDescription.pdf*** (Program logic description and output) files**.** Keep them in a folder. Then compress the folder and rename it with your registration number. You should submit this compressed file.

*****