CS1020 Sit-in Lab 01 A - Longest Substring

Semester 2 AY2013/2014

Welcome to the first sit-in lab!

Problem Description

Given a string composed of alphanumeric characters (lowercase and uppercase letters a..z, and numbers 0..9) as input, find and output the length of the longest substring (a consecutive block of characters) **without** any repeated characters in it. Uppercase and lowercase instances of the same letter are considered the same, that is, a = A, b = B etc. Note that the longest substring in a string may not be unique. Use Java String class to help you in this task (**not using String class will result in marks being halved!**).

Input

The given string. An example is given below:

```
ahhejjkaeK1i97ae
```

In the example given, there are three longest substrings without any repeated characters. They are ek1i97a, aek1i97 and K1i97ae, all having a length of 7. kaeK1i97 cannot be the longest substring according to our criteria since k is repeated.

Output

The length of the longest substring (**Not the substring itself !**). The output for the above input is given below:

```
7
```

Skeleton Program

Your program is to be named **LongestSubstring.java** (do not change this name). A skeleton program is provided, but you can change it any way or add new class(es) and method(s) as you deem fit. The skeleton program is as follows:

```
import java.util.*;

/* Getting length of longest substring without repeated characters in a string */
public class LongestSubstring {

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
  }
}
```

Testing your program

The following input and output files are in your plab account:

```
LongestSubstringTC1.in
LongestSubstringTC2.in
LongestSubstringTC1.out
LongestSubstringTC2.out
```

LongestSubstringTC1.in and LongestSubstringTC2.in are the input test cases, while LongestSubstringTC1.out and LongestSubstringTC2.out are the expected output for the respective test cases.

After you have compiled your program, to test it with say LongestSubstringTC1.in, you type:

```
java LongestSubstring < LongestSubstringTC1.in
```

Grading Scheme

1. Program correctness = 70 marks, Design = 20 marks, Programming = 10 marks
2. No marks awarded if the program does not compile.
3. Marks will be deducted if **student particulars and program description** are not filled up in the top portion of the source code.
4. There are 10 test cases. Each test case is worth 7 marks.
5. Marks will be halved if String class is not used.
6. Things to look out for under design
   (a) correct usage of programming constructs (eg use correct type for variables)
   (b) Not overly complicated logic
   (c) No redundant logic
7. Things to look out for under programming style
   (a) Meaningful comments (including pre- and post-condition description if neccessary)
   (b) Proper indentation
   (c) Meaningful identifiers