# Evaluation Rubric :

|  |  |  |
| --- | --- | --- |
| **Evaluation parameter** | **Does not meet specifications** | **Meets specifications** |
| **Problem statement** |  | **2** |
| Problem Statement must be clearly defined |  | **√** |
| Expected input and output formats must be described |  | **√** |
| Explain the problem statement with an example(if applicable) |  | **√** |
| **Expected input & output** |  | **3** |
| Minimum of 5 test cases (if applicable) |  | **√** |
| Coverage |  | **√** |
| Border condition |  | **√** |
| Unexpected inputs |  | **√** |
| **Solution** |  | **5** |
| The correctness of the solution. |  | **√** |
| Check for all the elements (tokens) of the problem (Assignment, Arithmetic, conditional, relational, input, output etc) |  | **√** |
| **Trace Table :** |  | **5** |
| Columns are variables, conditions, print statements |  | **√** |
| Order |  | **√** |
| Trace table for each function(If applicable) |  | **√** |
| labeling the columns |  | **√** |
| Coverage (conditions, iterations... etc) |  | **√** |
| **Final Result** |  | **2** |
| Executable File Submission |  | **√** |
| **Executable File** |  | **3** |
| Check with all test cases |  | **√** |

## 

**Problem Statement**: **(2 Marks)**

Find the Largest alphabetical sequence in a given string.

**Test cases: (3 Marks)**

|  |  |
| --- | --- |
| **Expected Input** | **Expected Output** |
| munny | mu |
| eat | at |
| fog | fo |
| hello | el |
| Hemanth | ant |

**Solution**: **(5 Marks)**

Step 1: START  
Step 2: Input string, s

Step 3: Convert to characters, a=ToCharacters(s)

Step 4: len=length(a)

Step 5: Initialize=1, cm=0,mci=0, c=0

Step 6: while i<=len-1, go to step 6.1

Step 6.1: If a[i]=a[i+1], go to step 6.2 else go to 6.3

Step 6.2: c=c+1

Step 6.3: c=0

Step 7: if cm<c, go to step 7.1  
Step 7.1: cm=c, go to step 7.2  
Step 7.2: mci=i, go to step 7.3  
Step 7.3: i=i+1, go to step 6  
Step 8: WRITE subset(tostring(a),mci-cm+1,cm+1),go to step 9  
Step 9: END

**Trace Table** : **(5 Marks)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **len** | **i** | **i<=len-1** | **a[i]<a[i+1]** | **c=c+1** | **cm<c** | **cm** | **mci** |
| Munny | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  | 2 | 1 | 0 |  | 0 |  |  |
|  |  | 3 | 1 | 0 |  | 0 |  |  |
|  |  | 4 | 1 | 0 |  | 0 |  |  |
|  |  | 5 | 0 |  |  |  |  |  |

subset(a,2,2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **len** | **i** | **i<=len-1** | **a[i]<a[i+1]** | **c=c+1** | **cm<c** | **cm** | **mci** |
| eat | 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
|  |  | 2 | 1 | 1 | 1 | 1 | 1 | 2 |
|  |  | 3 | 0 |  |  |  |  |  |

subset(a,2,2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **len** | **i** | **i<=len-1** | **a[i]<a[i+1]** | **c=c+1** | **cm<c** | **cm** | **mci** |
| fog | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  | 2 | 1 | 0 |  |  |  |  |
|  |  | 3 | 0 |  |  |  |  |  |

subset(a,1,2)

**Final Result :** **(2 Marks)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Expected input** | **Expected output** | **Actual output** | **Test result** |
| munny | mu | mu | 1 |
| eat | ea | ea | 1 |
| fog | fo | fo | 1 |