Evaluation Rubric :

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

**Problem Statement:** Display even characters from the given string

**Explanation:** The input string is used to predict index of a characters present in the String. Index is initially 0. We check if index is even for the given length of string using condition i/2=0. In case, index is odd . Increment the value of index when i/2= 1.

|  |  |
| --- | --- |
| **Expected Input** | **Expected Output** |
| A | A |
| Hello | H, l, o |
| R U | R U |
| I am | I a |
| It is | I i |

**Algorithm** To find even characters from the string

Step 1: Start

Step2: Read String s

Step 3: Convert String into individual characters using toCharacter and Output String

Step 4: Check for Length of the string, n= Length(s)

Step 5: Initially index of String i=0

Step 5.1: Check if i/2 =0 , Print s[i]

Step 6: Else increment i=i+1 , Go to Step 5.1

Step 7: If I > n . Condition is false , Go to Step 8

Step 8: End

**Trace table:**

Test Case 1: If String = ‘To’ , Length of String is even

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **i** | **I<n** | **I%2=0 (true)** | **I=i+1(false)** | **Result s[i]** |
| String s=To | 0 |  |  |  |  |
| Length(s)=2 | 0 | 0<2 | 0%2=0 |  | True S[0] |
|  | 1 | 1<2 | ½=0.5 | 1=i+1=2 | False |
|  | 2 | 2<2 | False |  | End IF |
|  |  |  |  |  | Print S[0] |
|  |  |  |  |  | O/p: ‘T’ |

Test Case 2: If String= ‘For’ , Length of String is Odd

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **i** | **I<n** | **I%2=0 (true)** | **I=i+1(false)** | **Result s[i]** |
| String s= For | 0 |  |  |  |  |
| Length(s)=3 | 0 | 0<3 | 0%2=0 |  | True S[0] |
|  | 1 | 1<3 | ½=0.5 | 1=i+1=2 | False |
|  | 2 | 2<3 | 2%2=0 | 2=i+1=3 | True S[2] |
|  | 3 | 3<3 | 3%2=1 | 3=i+1 | False |
|  | 4 | 4<3 |  |  | EndIF |
|  |  |  |  |  | Print S[0], S[2] |
|  |  |  |  |  | O/p: ‘F’, ‘r’ |

Test Case 3: If String = ‘A’ , Length of String is odd, Single character

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **i** | **I<n** | **I%2=0 (true)** | **I=i+1(false)** | **Result s[i]** |
| String s=A | 0 |  |  |  |  |
| Length(s)=1 | 0 | 0<1 | 0%2=0 | 0=i+1=1 | True S[0] |
|  | 1 | 1<2 | ½=0.5 | 1=i+1=2 | False |
|  | 2 | 2<2 | False |  | End IF |
|  |  |  |  |  | Print S[0] |
|  |  |  |  |  | O/p: ‘A’ |

Test Case 3: If String is ‘R U’, Length of string is 3 and has space

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **i** | **I<n** | **I%2=0 (true)** | **I=i+1(false)** | **Result s[i]** |
| String s=’R U’ | 0 |  |  |  |  |
| Length(s)=3 | 0 | 0<3 | 0%2=0 | 0=i+1=1 | True S[0] |
|  | 1 | 1<3 | ½=0.5 | 1=i+1=2 | False |
|  | 2 | 2<3 | 2%2=0 | I=2+1=3 | True S[2] |
|  | 3 | 3<2 | False | False | Print S[0], S[2] |
|  |  |  |  |  | O/p: ‘R’, ‘U’ |

**Final Result :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Expected input** | **Expected output** | **Actual output** | **Test result** |
|  |  |  |  |
| A | A | A | Prints: S[0] |
| Hello | H, l, o | H, l, o | Prints: S[0], S[2],S[4] |
| R U | R U | R U | Prints: S[0] |
| I am | I a | I a | Prints: S[0], S[4] |
| It is | I i | I i | Prints: S[0],S[2],S[4] |