# 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**: **(2 Marks)**

Check whether the given number is Armstrong or not

Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

**Test cases: (3 Marks)**

|  |  |
| --- | --- |
| **Expected Input** | **Expected Output** |
| 2 | 1 |
| 153 | 1 |
| 100 | 0 |
| -1 | Invalid |
| 0 | Invalid |

**Solution**:

**(5 Marks)**

Step 1: START

Step 2: Input number n  
 Step 2.1: if n<=0, print “Invalid” and go to Step 5

Step 2.2: sum=0 , temp=n

Step 3: If n>0 go to Step 3.1 else step 4

Step 3.1: r=n%10

sum=sum+(r\*r\*r);

n=floor(n/10); go to step 3

Step 4: if (temp=n) go to step 4.1

Step 4.1: Print Armstrong number else Print not a armstrong number go to step 5

Step 5: End

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

Test case 1: n=2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | r=n%10 | sum = sum+cube(r) | n>=0 | n/=10 | temp=n |
| 2 | 0 | 0 | 0 | 0 | TRUE |
|  |  |  |  |  |  |

Test case 2: n=131

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | r=n%10 | sum = sum+cube(r) | n>=0 | n/=10 | temp=n |
| 131 | 1 | 0+1 | 1 | 13 |  |
| 13 | 3 | 1+(cube 3) | 1 | 1 |  |
| 1 | 1 | 11 | 1 | 0 | FALSE |

Test case 3: n=153

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | r=n%10 | sum = sum+cube(r) | n>=0 | n/=10 | temp=n |
| 153 | 1 | 1 | 1 | 15 |  |
| 15 | 5 | 126 | 1 | 5 |  |
| 1 | 3 | 153 | 1 | 0 | TRUE |

Pseudocode:

START

READ n

res=0

o=n

WHILE o >=1 DO

rem=o%10

res = res + (rem\*rem\*rem)

o = o/10

ENDWHILE

IF n= res THEN

WRITE 'false'

ELSE

WRITE 'true'

ENDIF

END

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Expected input** | **Expected output** | **Actual output** | **Test result** |
| 2 | 1 | Armstrong | True |
| 153 | 1 | Armstrong | True |
| 100 | 0 | Not Armstrong | False |