Valid IP Address

ZPRAC-16-17-LabExam-2_Session-1

Valid IP Address [30*Marks*]

ANNOUNCEMENT: Up to 20% marks will be allotted for good programming practice. These include
- Comments for non trivial code
- Indentation: Align your code properly
- Meaningful variable names

A valid IP address must be in the form of A.B.C.D, where A, B, C and D are numbers from 0 to 255 (0 and 255 inclusive). Also, the non-zero numbers from A, B, C, D cannot have leading zeros (e.g. 007 is invalid). If A, B, C or D are zero, then it should compose of only one 0 (e.g. 00 or 000 are invalid). See examples below for better understanding.

Given a string as input, check whether the strings forms a valid IP address or not. If yes, print the sum (A+B+C+D). If no, print "Not Valid".

Example of some valid strings as IP addresses:
"1.2.3.4"
"225.12.0.0"
"1.0.2.2"
"0.0.0.0"

Example of some invalid strings as IP addresses:
"1234.1.2.3" -- 1234 is greater than 255
"1.02.1.3" -- 02 contains a leading zero
"1.000.2.2" -- 000 contains two leading zeros
"1.0..1.2" -- invalid dots

Input:
In the first line of input, an integer t is given, denoting the number of test cases. ($0 \le t \le 10^3$).
In each of the next t lines, a string s is given to check for a valid IP address. (size of $s \le 100$).

Input string can only contain characters from '0' to '9' and '.'.

Output:
For each of the *t* strings, print the desired output as mentioned above in a newline.

Note: Use the template provided, complete the defined ValidIp(.) function.

Example:
Input -

```
7
1.2.3.4
4294967299.1.1.1
0.1.0.1
12.2.01.1
255.255.255.255
0.00.12.2
256.12.1.1
```

Output -

```
10
Not Valid
2
Not Valid
1020
Not Valid
Not valid
```