Return-Codes

May 18, 2020

0.0.1 Problem statement

In an encryption system where ASCII lower case letters represent numbers in the pattern `a=1`, `b=2`, `c=3...` and so on, find out all the codes that are possible for a given input number.

Example 1

- `number = 123`
- `codes_possible = ["aw", "abc", "lc"]`

Explanation: The codes are for the following number:

- 1 . 23 = "aw"
- 1 . 2 . 3 = "abc"
- 12 . 3 = "lc"

Example 2

- `number = 145`
- `codes_possible = ["ade", "ne"]`

Return the codes in a list. The order of codes in the list is not important.
Note: you can assume that the input number will not contain any 0s

```
In [1]: def all_codes(number):
            """
            :param: number - input integer
            Return - list() of all codes possible for this number
            TODO: complete this method and return a list with all possible codes for the input n
            """
            pass
```

Show Solution

```
In [6]: def test_function(test_case):
            number = test_case[0]
            solution = test_case[1]

            output = all_codes(number)
```

```
            output.sort()
            solution.sort()

            if output == solution:
                print("Pass")
            else:
                print("Fail")
```

```
In [9]: number = 123
        solution = ['abc', 'aw', 'lc']
        test_case = [number, solution]
        test_function(test_case)
```

```
Pass
```

```
In [12]: number = 145
         solution =  ['ade', 'ne']
         test_case = [number, solution]
         test_function(test_case)
```

```
Pass
```

```
In [11]: number = 1145
         solution =  ['aade', 'ane', 'kde']
         test_case = [number, solution]
         test_function(test_case)
```

```
Pass
```

```
In [13]: number = 4545
         solution = ['dede']
         test_case = [number, solution]
         test_function(test_case)
```

```
Pass
```