Decode Ways

Question: A message containing letters from A-Z is being encoded to numbers using the following mapping:

'A' -> 1

'B' -> 2

...

'Z' -> 26

Given an encoded message containing digits, determine the total number of ways to decode it.

For example:

Given encoded message "12", it could be decoded as "AB" (1 2) or "L" (12).

The number of ways decoding "12" is 2.

Solutions:

```
class Solution:
    # @param s is a string
    # @return an integer
    def numDecodings(s):
        if s=="" or s[0]=='0': return 0
        dp=[1,1]
        for i in range(2,len(s)+1):
            if 10 <=int(s[i-2:i]) <=26 and s[i-1]!='0':
                dp.append(dp[i-1]+dp[i-2])
            elif int(s[i-2:i])==10 or int(s[i-2:i])==20:
                dp.append(dp[i-2])
            elif s[i-1]!='0':
                dp.append(dp[i-1])
            else:
                return 0
        return dp[len(s)]

Solution.numDecodings("12")
```