

Cryptarithmic

1. You are given three strings s1, s2 and s3.
2. First two are supposed to add and form third. $s1 + s2 = s3$
3. You have to map each individual character to a digit, so that the above equation holds true.

Note -> Check out the question video and write the recursive code as it is intended without changing the signature. The judge can't force you but intends you to teach a concept.

Input Format

Three strings

s1

s2

s3

$1 \leq \text{length of } s1, s2, s3 \leq 10$

Sample Input

team

pep

toppr

Sample Output

{'t': 0, 'm': 2, 'r': 5, 'p': 3, 'o': 1, 'a': 4, 'e': 9}

{'t': 0, 'm': 2, 'r': 6, 'p': 4, 'o': 1, 'a': 5, 'e': 9}

{'t': 0, 'm': 2, 'r': 7, 'p': 5, 'o': 1, 'a': 6, 'e': 9}

{'t': 0, 'm': 2, 'r': 8, 'p': 6, 'o': 1, 'a': 7, 'e': 9}

{'t': 0, 'm': 3, 'r': 7, 'p': 4, 'o': 1, 'a': 5, 'e': 9}

{'t': 0, 'm': 3, 'r': 8, 'p': 5, 'o': 1, 'a': 6, 'e': 9}

{'t': 0, 'm': 4, 'r': 6, 'p': 2, 'o': 1, 'a': 3, 'e': 9}

{'t': 0, 'm': 5, 'r': 7, 'p': 2, 'o': 1, 'a': 3, 'e': 9}

{'t': 0, 'm': 5, 'r': 8, 'p': 3, 'o': 1, 'a': 4, 'e': 9}

{'t': 0, 'm': 6, 'r': 8, 'p': 2, 'o': 1, 'a': 3, 'e': 9}

```

def cryptArithmetic(s1,s2,s3):
    unique = ''.join(list(set(s1+s2+s3)))
    fmap = {}
    visited = [False]*10
    cryptArithmeticSolver(unique,0,fmap,s1,s2,s3,visited)
    return

def cryptArithmeticSolver(unique,idx,fmap,s1,s2,s3,visited):
    if idx==len(unique):
        n1 = getNumber(fmap,s1)
        n2 = getNumber(fmap,s2)
        n3 = getNumber(fmap,s3)
        if n1+n2==n3:
            print(fmap)
            return

    for i in range(10):
        if visited[i]==False:
            visited[i]=True
            fmap[unique[idx]]=i
            cryptArithmeticSolver(unique, idx+1, fmap, s1, s2, s3, visited)
            visited[i] = False
            del fmap[unique[idx]]

def getNumber(fmap,s):
    string = ''
    for ch in s:
        string+=str(fmap[ch])
    return int(string)

cryptArithmetic('team','pep','toppr')

```