Coin Change - Permutations - 1

1. You are given a number n, representing the count of coins.
2. You are given n numbers, representing the denominations of n coins.
3. You are given a number "amt".
4. You are required to calculate and print the permutations of the n coins (non-duplicate) using which the amount "amt" can be paid.
Note -> Use the code snippet and follow the algorithm discussed in question video. The judge can't force you but the intention is to teach a concept. Play in spirit of the question.

A number n
n1
n2
.. n number of elements
A number amt

1 <= n <= 30
0 <= n1, n2, .. n elements <= 20
0 <= amt <= 50

5
2
3
5
6
7
12

Sample Output

2-3-7-.
2-7-3-.
3-2-7-.
3-7-2-.
5-7-.
7-2-3-.
7-3-2-.
7-5-.

```
def coinsChangePermutationI(coins,amount):
    idx = 0
    ans = []
```

```
    ssf=''
    visited = [False]*len(coins)
    helper(coins,amount,ans,ssf,visited)
    return ans

def helper(coins,amount,ans,ssf,visited):
    if amount==0:
        ans.append(ssf[:-1])
        return
    if amount<0:
        return
    for i in range(len(coins)):
        if visited[i]==False:
            visited[i]=True
            helper(coins, amount-coins[i],ans, ssf+str(coins[i])+'-',
visited)
            visited[i]=False


print(coinsChangePermutationI([2,3,5,6,7],15))
```