Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
4790508	12:19:30	Virus Replication	✔ **Accepted**	0.57 s	Python 3
	✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔ ✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
virus_260621270.py	574 bytes	c18a5f96d9e6934fafe903208156955f417be81d	download

Edit and resubmit this submission.

virus_260621270.py

```
# before is the first line
# after is the second line
# The program starts trimming from the front it stops when a char in the first
# string doesn't correspond with the second string. Then it does the same
# thing from the reverse and returns the length of the remainder or
# the virus itself.

before = input()
after = input()

while(len(before)!=0 and len(after)!=0 and before[0] == after[0]):
  before = before[1:]
  after = after[1:]

while(len(before)!=0 and len(after)!=0 and before[-1] == after[-1]):
  before = before[:-1]
  after = after[:-1]

print(len(after))
```

Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
4803501	16:02:46	Restaurant Orders	✔ **Accepted**	0.17 s	Python 3
	✓✓✓✓✓✓✓✓✓✓✓✓				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
orders_260621270.py	1324 bytes	af6f19867cb432a21928ffc64a3cbdd97e6b114c	download

Edit and resubmit this submission.

orders_260621270.py

```
# Coin Change as a 2D DP
# https://www.geeksforgeeks.org/coin-change-dp-7/
# COMP321-Lecture5-DP Solution Idea #1:
def count(S, m, n):
  if (n == 0):
    return 1
  if (n < 0):
    return 0
  if (m <=0 and n >= 1):
    return 0
  return count(S, m - 1, n) + count(S, m, n - S[m - 1])

# Top-Down DP
# COMP321-Lecture5-DP Solution Idea #2:
table = [0]
def topDownDP(numItems, itemCosts, numOrders, orderPrices):
  for i in range(max(orderPrices) + max(itemCosts)):
    table.append(-1)
  for i in range(numItems):
    for j in range(max(orderPrices) + 1):
      if table[j] == -2:
        table[j + itemCosts[i]] = -2
      if table[j] >= 0:
        if table[j + itemCosts[i]] == -1:
          table[j + itemCosts[i]] = i
        else:
          table[j + itemCosts[i]] = -2

numItems = int(input())
itemCosts = list(map(int, input().split(" ")))
numOrders = int(input())
orderPrices = list(map(int, input().split(" ")))

topDownDP(numItems, itemCosts, numOrders, orderPrices)

for total in orderPrices:
  if(table[total] == -1):
    print("Impossible")
  elif(table[total] == -2):
    print("Ambiguous")
  else:
    order = []
    while total > 0:
      order.append(table[total] + 1)
      total = total - itemCosts[table[total]]
    if total < 0:
      print("Ambiguous")
    else:
      print(*sorted(order), sep = ' ')
```