最长的顺子

题目描述：斗地主起源于湖北十堰房县，据传是一位叫吴修全的年轻人根据当地流行的扑克玩法"跑得快"改编的，如今已风靡整个中国，并流行于互联网上。
牌型:
单顺, 又称顺子，最少5张牌，最多12张牌（3…A），不能有2，也不能有大小王，不计花色
例如：3-4-5-6-7-8，7-8-9-10-J-Q，3-4-5-6-7-8-9-10-J-Q-K-A
可用的牌 3<4<5<6<7<8<9<10<J<Q<K<A<2 < B(小王)< C(大王)，每种牌除大小王外有4种花色(共有 13X4 + 2 张牌)
输入1. 手上已有的牌 2. 已经出过的牌（包括对手出的和自己出的牌）
输出: 对手可能构成的最长的顺子(如果有相同长度的顺子, 输出牌面最大的那一个)，如果无法构成顺子, 则输出 NO-CHAIN

输入描述：输入的第一行为当前手中的牌
输入的第二行为已经出过的牌

输出描述：最长的顺子

补充说明：

示例1
输入：3-3-3-3-4-4-5-5-6-7-8-9-10-J-Q-K-A
4-5-6-7-8-8-8
输出：9-10-J-Q-K-A
说明：

示例2
输入：3-3-3-3-8-8-8-8
K-K-K-K
输出：NO-CHAIN
说明：剩余的牌无法构成顺子

```
str1=list(input().split('-'))
str2=list(input().split('-'))
dic_card={}
card=['3','4','5','6','7','8','9','10','J','Q','K','A']
for i in card:
    dic_card[i]=4
for i in str1:
  if i!='2' and i !='B' and i!='C':
    if i in dic_card:
        dic_card[i]-=1
for i in str2:
 if i!='2' and i !='B' and i!='C':
    if i in dic_card:
        dic_card[i]-=1
s_card=[]
for i in dic_card:
    if dic_card[i]>0:
        s_card.append(i)
for i in range(len(s_card)):
    if s_card[i]=='J':
        s_card[i]=11
    elif s_card[i]=='Q':
```

```
            s_card[i]=12
        elif s_card[i]=='K':
            s_card[i]=13
        elif s_card[i]=='A':
            s_card[i]=14
s_card=sorted([int(i) for i in s_card])
s=[]
n=0
while n<len(s_card):
        count=1
        m=0
        for i in range(n,len(s_card)-1):
                if s_card[i+1]-s_card[i]==1:
                    count+=1
                    m=max(m,count)
                elif s_card[i+1]-s_card[i]!=1:
                     break
        s.append((n,m))
        n=n+1
s.sort(key=lambda x:(-x[1],-x[0]))
if s and s[0][1]>=5 :
    str3=''

    for i in range(s[0][0],s[0][0]+s[0][1]):
        if s_card[i]==11:
                str3+='J-'
        elif s_card[i]==12:
                str3+='Q-'
        elif s_card[i]==13:
                str3+='K-'
        elif s_card[i]==14:
                str3+='A-'
        else:
                str3+=str(s_card[i])+'-'
    print(str3[:-1])
else:
        print('NO-CHAIN')
```