isomorphic strings

```
class Solution:
    def isIsomorphic(self, s: str, t: str) -> bool:
        if len(s)!=len(t):
            return False
        x={}
        y={}
        for a,b in zip(s,t):
            if a in x and x[a]!=b:
                return False
            if b in y and y[b]!=a:
                return False
            x[a]=b
            y[b]=a
        return True
```

reverse linked list

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reverseList(self, head: Optional[ListNode]) -> Optional[ListNode]:
        if not head or not head.next :
            return head
        stack=[]
        while head.next :
            stack.append(head)
            head=head.next
        cur=head
        while stack :
            cur.next=stack.pop()
            cur=cur.next
        cur.next=None
        return head
```