13. Perform string reversal using stack

The idea is to create any empty stack and push all the character from the string into it. Then pop each character one by one from the stack and put them back into the input string starting from 0th index.
Stack work on the principle of FILO (First in last out)
After popping all the elements and placing them back to string the formed string would be reversed

Algorithm

Step 1: Accept string from the user
Step 2: Scan complete string from left to right.
Step 3: push each element character in stack
Step 4: Once scanning is done
Step 5: Start popping element if to recall 1st from stack and append it to result string
Step 6: Once the stack is empty return the result in string from which is the final result
Step 7: Display the result

Time complexity - O(n)
Space Complexity - O(n)

Example

String SONU

	String	operation	stack()
1.	S	push S in stack	S
2.	O	push O	S,O
3.	N	push N	S,O,N
4.	U	push U	S,O,N,U

∴ Stack is

U
N
O
S

Now string becomes = UNOS after popping one after element from stack.

Code:-

```
class stack:
    def __init__(self):
        self.stack=[]
    def is_empty(self):
        return len(self.stack)==0
    def push(self,item):
        self.stack.append(item)
    def pop(self):
        if len(self.stack)==0
            return "stack is empty"
        return self.stack.pop()
    def peak(self):
        return self.stack[-1]
    def display(self):
        print(self.stack)
input_string>input()
s=stack()
for character in input_string:
    s.push(character)
output_string=" "
for i in range(len(s.stack))
    output_string+=s.pop()
print(output_string)
```

Output:

Sony
unos