Stack Assignment

1) WAP to implement a stack of marks using user defined function :- ~~Trade by Python Tutor, Inc~~
   a) PUSH() to push marks of a student in a stack.
   b) POP() an element from the stack.
   c) Display the stack of marks.

```
Stack = []
top = -1

def push_Stack():
    global top
    m = int(input('Enter marks to push:'))
    top = top + 1
    Stack.append(m)
    print(m,'is pushed in stack')
def pop_Stack():
    global top
    if len(Stack)==0:
        print('memory underflow / stack is empty')
    else:
        m = Stack.pop()
        print('elements deleted from the top of stack',m)
        top = top - 1
def display():
    global top
    if Stack=[]
        print('stack is empty or memory underflow
              so nothing to display')
```

my-jor-id()

```
    else:
        print('top->', Stack[top])
        for i in range(top-1,-1,-1):
            print(Stack[i])

while True:
    print('1   for push in stack')
    print('2   for pop in stack')
    print('3   for display in stack')
    choice= int(input('enter your choice:'))
    if choice==1:
        push_Stack()
        print('top is at', top)
    elif choice==2:
        pop_Stack()
        print('top is at', top)
    elif choice==3:
        display()
    else:
        print('wrong choice')
```

Q WAP to reverse a string using PUSH and POP operation in Stack

```
stack=[]
stack_new=[]
top=-1

def push():
    global top
    a=input('enter a sentence:')
    for i in a:
```

```
    Stack.append(i)
def pop():
    global top
    for i in range(len(Stack)):
        m= Stack.pop()
        top= top-1
        Stack-new.append(m)
def display():
    global top
    m= ''.join(stack_new)
    print('String in reverse:', m)

push()
pop()
display()
```

3) WAP to create a Stack for storing only odd numbers out of all the numbers entered by the user. Display the contents of the Stack along with the largest odd number in the stack.

```
Stack = []
top= -1

def push():
    global top
    while True
    num= int(input('Enter a number:'))
    if num%2 !=0:
        Stack.append(num)

    choice = input('Do you want to enter more(Y/N)')
```

```
if choice in 'Nn':
    top = len(stack) - 1
    break.

def pop():
    max = min(stack)
    if len(stack) == 0:
        print('Underflow: stack is empty')
    else:
        for i in range(len(stack)):
            num = stack.pop()
            if num > max:
                max = num
    print('\n The maximum number in the stack is :', max)

push()
display()
pop()
```