

MidTerm1_4031_Q2

April 10, 2025

```
[17]: # Q1
class Queue:
    def __init__(self):
        self.data=[]
    def is_empty(self):
        return len(self.data)==0
    def push(self,x):
        self.data.append(x)
    def pop(self):
        if self.is_empty()==True:
            return 0,None
        else:
            y=self.data[0]
            self.data.pop(0)
            return 1,y
    def first_element(self):
        if self.is_empty()==True:
            return 0,None
        else:
            y=self.data[0]
            return 1,y

    def show(self):
        if self.is_empty():
            print('Queue is empty')
        else:
            for x in self.data:
                print(x,end='->')
            print()

# q=Queue()
# # print(q.is_empty())

# for i in range(5):
#     a=float(input('enter a real number: '))
#     q.push(a)
```

```

#     q.show()
# # # print(q.is_empty())
# # while not q.is_empty():
# #     q.pop()
# #     q.show()
# # # print(q.is_empty())
# out=q.first_element()
# if out[0]==1:
#     print(out[1])

# out=q.pop()
# if out[0]==1:
#     print(out[1])
#     q.show()
# else:
#     print('Queue is empty()')

# flag,val=q.pop()
# if flag==1:
#     print(val)
#     q.show()
# else:
#     print('Queue is empty()')

```

```

[20]: # Q2
class Stack:
    def __init__(self,n):
        self.N=n
        self.data=[None]*n
        self.top=-1

    def is_empty(self):
        return self.top== -1

    def is_full(self):
        return self.top==self.N-1

    def push(self,x):
        if not self.is_full():
            self.top+=1
            self.data[self.top]=x
            return 1
        else:
            return 0

    def pop(self):

```

```

        if not self.is_empty():
            y=self.data[self.top]
            self.top-=1
            return 1,y
        else:
            return 0,None

    def top_element(self):
        if not self.is_empty():
            y=self.data[self.top]
            return 1,y
        else:
            return 0,None

    def show(self):
        if not self.is_empty():
            for i in range(self.top+1):
                print(self.data[i],end='->')
            print()
        else:
            print('Stack is empty')

# s=Stack(10)
# print(s.is_empty())

# for i in range(5):
#     a=float(input('enter a real number: '))
#     s.push(a)
#     s.show()
# print(s.is_empty())
# # while not s.is_empty():
# #     s.pop()
# #     s.show()
# # print(s.is_empty())
# out=s.top_element()
# if out[0]==1:
#     print(out[1])

# out=s.pop()
# if out[0]==1:
#     print(out[1])
#     s.show()
# else:
#     print('Stack is empty()')

```

```
# flag, val=s.pop()
# if flag==1:
#     print(val)
#     s.show()
# else:
#     print('Stack is empty()')
```

[]:

[]:

[]:

[]: