

# SaftakandEstaf

December 9, 2022

## 1 Saftak and Estaf

### 1.1 Implement a queue with a stack and implement a stack with a queue

#### 1.1.1 A queue implemented with a stack

```
...
datatype: StackBaseQueue
methods: appendleft, pop
attributes: underlying_stack
...

- Documentation of `appendleft`:

...
appendleft:
# Appends a new entry to queue.
# It is equivalence of pushing up to the underlying_stack
# e.g. appendleft(6):

-----
| 5 | 4 | 3 | 2 | 1 |
-----

-----
-> | 6 | 5 | 4 | 3 | 2 | 1 |
-----
...

- Implementation of `appendleft`:

...
appendleft(self, data):
    self.underlying_stack.push(data)
...

- Documentation of `pop`:

...
```

```

pop:
# pop return the first object in the queue with it is its turn to be popped.
# e.g. pop()

```

```

-----
| 6 | 5 | 4 | 3 | 2 | 1 | ->
-----

```

```

returned: 1

```

```

-----
| 6 | 5 | 4 | 3 | 2 |
-----
...

```

- Implementation of `pop`:

```

...
pop(self):
    temp = [] # a stack
    while self.underlying_stack: # iterates until the stack is empty
        temp.push(self.underlying_stack.pop())

    # temp: [1, 2, 3, 4, 5, 6]
    to_return = temp.pop()

    # temp: [2, 3, 4, 5, 6]
    while temp:
        self.underlying_stack.push(temp.pop())

    # underlying_stack: [6, 4, 3, 2, 1]

    return to_return
...

```

---

### 1.1.2 A stack implemented with a queue

```

...
datatype: QueueBaseStack
methods: push, pop
attributes: underlying_queue
...

```

- Documentation of `push`:

```

...
push:
# Pushing items up to the stack is done easily with
# `appendleft` method of underlying_queue
# e.g. push(6)

```

```

-----
| 5 | 4 | 3 | 2 | 1 |
-----

```

```

-----
-> | 6 | 5 | 4 | 3 | 2 | 1 |
-----

```

```

...

```

- Implementation of `push`:

```

...
push(self, data):
    self.underlying_queue.appendleft(data)
...

```

- Documentation of `pop`:

```

...
pop:
# Pops the element which has been added recently
# e.g. pop()

```

```

-----
| 6 | 5 | 4 | 3 | 2 | 1 |
-----

```

returned value: 6

```

-----
| 5 | 4 | 3 | 2 | 1 |
-----

```

```

...

```

- Implementation of `pop`:

```

...
pop(self):
    temp = [] # a queue
    while self.underlying_queue:
        temp.appendleft(self.underlying_queue.pop())

```

```

    # temp: [1, 2, 3, 4, 5, 6]
    to_return = temp.pop()

    while temp:
        self.underlying_queue.appendleft(temp.pop())

    # underlying_queue: [1, 2, 3, 4, 5]

    return to_return
'''

```

### 1.1.3 Notes:

1. Because `appendleft`, `pop`, `push` and `pop` are instance methods of their `datatype` they must have at least `self` parameter
2. Each code segment start with whether a backtick or three backticks.