

What is a Stack (LIFO)?

A **stack** is a linear data structure where elements are **added (pushed)** and **removed (popped)** from the **same end**, called the **top** of the stack.

Common Stack Operations:

Operation	Description
<code>push(x)</code>	Adds element <code>x</code> to the top
<code>pop()</code>	Removes the top element
<code>peek()</code> or <code>top()</code>	Returns the top element without removing it
<code>isEmpty()</code>	Checks if the stack is empty

```
class Stack:
```

```
    def __init__(self):
```

```
        self.items = []
```

```
    def is_empty(self):
```

```
        return len(self.items) == 0
```

```
    def push(self, item):
```

```
        self.items.append(item)
```

```
    def pop(self):
```

```
        if not self.is_empty():
```

```
            return self.items.pop()
```

```
        else:
```

```
            print("Stack is empty")
```

```
            return None
```

```
    def peek(self):
```

```
if not self.is_empty():  
    return self.items[-1]  
  
else:  
    print("Stack is empty")  
    return None
```

```
def size(self):  
    return len(self.items)
```

Example usage:

```
my_stack = Stack()  
my_stack.push(10)  
my_stack.push(20)  
print(my_stack.items)  
print(f"Top element: {my_stack.peek()}")  
print(f"Popped element: {my_stack.pop()}")  
print(f"Stack size: {my_stack.size()}")  
print(f"Popped element: {my_stack.pop()}")  
print(f"Stack size: {my_stack.size()}")  
print(f"Stack size: {my_stack.is_empty()}")
```

#same example you can try with deque method.

#Hint - from collections import deque