

Day-13 : Stack & Queue

Problem 1: Implement a stack using an array.

```
class Stack:
```

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

```
        self.stack = []
```

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

```
        self.stack.append(item)
```

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

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

```
            return self.stack.pop()
```

```
        else:
```

```
            return None
```

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

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

```
    def print_stack(self):
```

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

```
            print("Stack:")
```

```
            for item in reversed(self.stack):
```

```
                print(item)
```

```
        else:
```

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

```
stack = Stack()
```

```
stack.push(10)
```

```
stack.push(20)
```

```
stack.push(30)
```

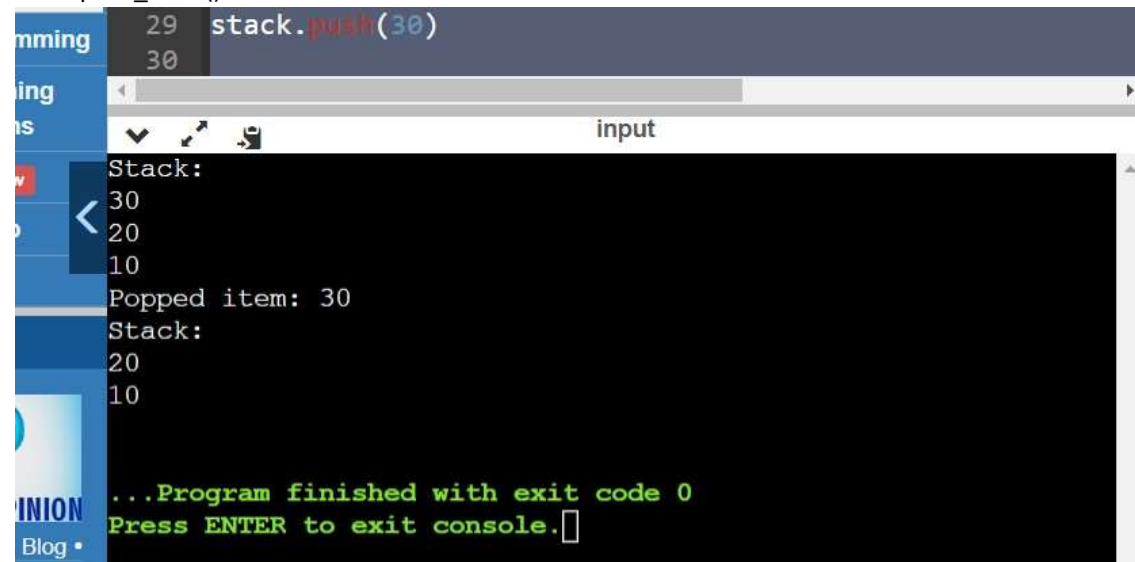
```
stack.print_stack()
```

```
popped_item = stack.pop()
```

```
if popped_item is not None:
```

```
    print("Popped item:", popped_item)
```

```
stack.print_stack()
```



The screenshot shows a code editor on the left with a dark theme. The code in the editor is as follows:

```
29 stack.push(30)
30
```

Below the code editor is a terminal window with a black background and green text. The terminal output is:

```
Stack:
30
20
10
Popped item: 30
Stack:
20
10
...Program finished with exit code 0
Press ENTER to exit console.
```

The terminal window has a title bar that says "input". On the left side of the terminal window, there is a vertical sidebar with a blue background and white text that says "UNION Blog".