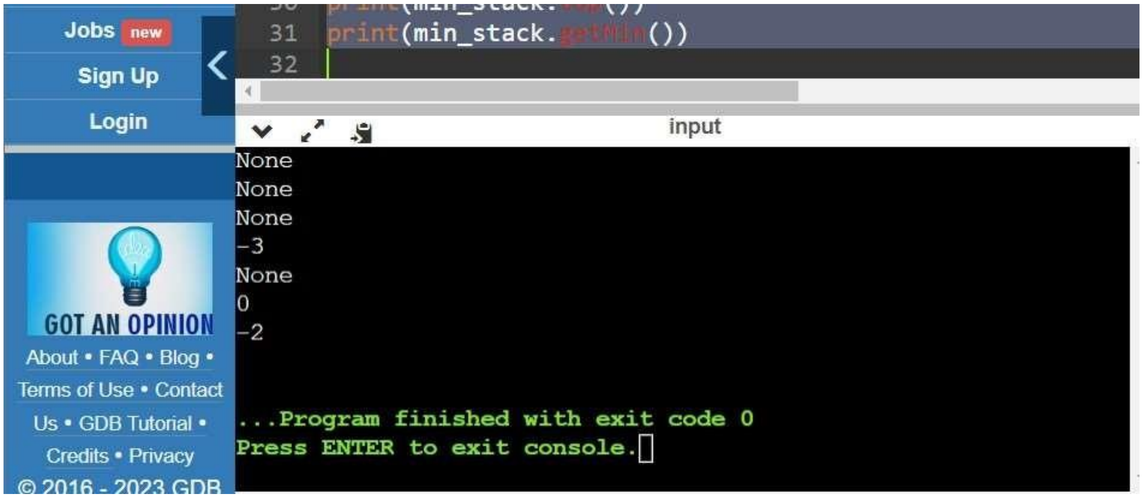
Tanish Mittal
IT-C
2000290130174

Problem 5: Implement Min Stack | O(2N) and O(N) Space Complexity.Design astack that supports push, pop, top, and retrieving the minimum element in constant time.

```
class MinStack:

    def___init_(self):

        self.stack = []

        self.min_stack = []


    def push(self, val):

        self.stack.append(val)

        if not self.min_stack or val <= self.min_stack[-1]:

            self.min_stack.append(val)


    def pop(self): if
        self.stack:

            val = self.stack.pop()

            if val == self.min_stack[-1]:

                self.min_stack.pop()


    def  top(self): if

        self.stack:

            return self.stack[-1]


    def getMin(self):

        if self.min_stack:
```

```
        return self.min_stack[-1]
min_stack = MinStack()
print(min_stack.push(-2))
print(min_stack.push(0))
print(min_stack.push(-3))
print(min_stack.getMin())
print(min_stack.pop())
print(min_stack.top())
print(min_stack.getMin())
```



```
print(min_stack.getMin())
```

```
None
None
None
-3
None
0
-2


...Program finished with exit code 0
Press ENTER to exit console.
```