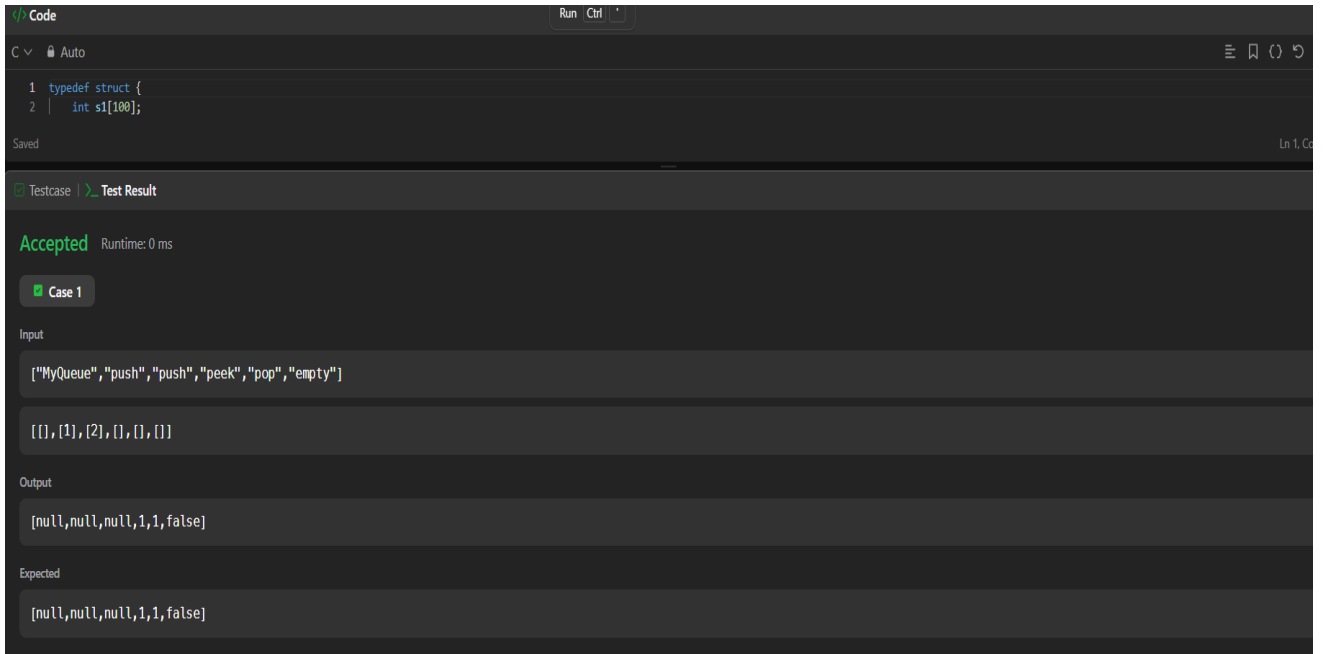


Implement Queue Using Stacks



The screenshot shows a code editor with a C++ implementation of a queue using stacks. The code defines a struct with an array of integers and implements push, pop, peek, and empty methods. A test case is shown with the input ["MyQueue", "push", "push", "peek", "pop", "empty"] and the expected output [null, null, null, 1, 1, false]. The test result is "Accepted".

```
1 typedef struct {
2     int s1[100];
3 } MyQueue;
4
5 void MyQueue_push(MyQueue* obj, int x) {
6     obj->s1[obj->size] = x;
7     obj->size++;
8 }
9
10 int MyQueue_pop(MyQueue* obj) {
11     if (obj->size > 0) {
12         obj->size--;
13         return obj->s1[obj->size];
14     }
15     return -1;
16 }
17
18 int MyQueue_peek(MyQueue* obj) {
19     if (obj->size > 0) {
20         return obj->s1[obj->size - 1];
21     }
22     return -1;
23 }
24
25 bool MyQueue_empty(MyQueue* obj) {
26     return obj->size == 0;
27 }
```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1

Input

["MyQueue", "push", "push", "peek", "pop", "empty"]

[[], [1], [2], [], [], []]

Output

[null, null, null, 1, 1, false]

Expected

[null, null, null, 1, 1, false]