

implement Stack using Queue

```
void QueueStack::push(int x) {
    // Push element into q2
    q2.push(x);

    while (!q1.empty()) {
        q2.push(q1.front());
        q1.pop();
    }

    swap(q1, q2);
}

// Function to pop an element from stack using two queues.
int QueueStack::pop() {
    if (q1.empty()) return -1;

    int topElement = q1.front();
    q1.pop();
    return topElement;
}
```

The screenshot shows a C++ IDE interface. On the left, the 'Output Window' is open, displaying 'Compilation Results' for a problem titled 'Queue using two Stacks'. The results indicate 'Problem Solved Successfully' with 100/100 test cases passed, 2/2 points scored, and a time taken of 0.01. The right pane shows the C++ code for the QueueStack class, which implements the push and pop operations using two queues (q1 and q2). The code is as follows:

```
1 // Driver Code Ends
2
3 /* The structure of the class is
4 * class QueueStack{
5 * private:
6 *     queue<int> q1;
7 *     queue<int> q2;
8 * public:
9 *     void push(int);
10 *     int pop();
11 * };
12 */
13
14 void QueueStack::push(int x) {
15     q2.push(x);
16
17     while (!q1.empty()) {
18         q2.push(q1.front());
19         q1.pop();
20     }
21
22     swap(q1, q2);
23 }
24
25 // Function to pop an element from stack using two queues.
26 int QueueStack::pop() {
27     if (q1.empty()) return -1;
28
29     int topElement = q1.front();
30     q1.pop();
31     return topElement;
32 }
```