

AP Assignment – 6

Name : Vipul

UID : 22BCS16120

Ques . 1 Implement Stack using an Array

Problem link : <https://www.geeksforgeeks.org/problems/implement-stack-using-array/1>

The screenshot displays the GeeksforGeeks problem-solving interface. On the left, the 'Output Window' shows 'Compilation Results' for the problem 'Y.O.G.I. (AI Bot)'. It indicates 'Problem Solved Successfully' with a green checkmark. The 'Test Cases Passed' are 1115 / 1115. The 'Attempts : Correct / Total' are 1 / 1. The 'Accuracy : 100%'. The 'Points Scored' are 1 / 1, and the 'Your Total Score' is 30. The 'Time Taken' is 0.02. Below this, the 'Solve Next' section offers links to 'Stack using Linked List', 'Queue Using Array', and 'Pairwise Consecutive Elements'. On the right, the code editor shows a C++ solution for implementing a stack using an array. The code defines a class 'MyStack' with a private array 'arr' of size 1000 and a 'top' pointer. It includes methods 'push(int x)' and 'pop()'.

```
1 // } Driver Code Ends
19
20 // Function to push an integer into the stack.
21
22 /*
23 class MyStack
24 {
25 private:
26     int arr[1000];
27     int top;
28 public:
29     MyStack(){top=-1;}
30     int pop();
31     void push(int);
32 };
33 */
34
35 void MyStack ::push(int x) {
36     arr[++top]=x;
37 }
38
39 // Function to remove an item from top of the stack.
40 int MyStack ::pop() {
41     if(top==1)
42         return -1;
43     return arr[top--];
44 }
45
46 // } Driver Code Ends
47
```

Ques . 2 Implement Queue using an Array

Problem link : <https://www.geeksforgeeks.org/problems/implement-queue-using-array/1>

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓ [Suggest Feedback](#)

Test Cases Passed: **170 / 170**

Attempts: Correct / Total: **1 / 2**
Accuracy: 50%

Points Scored: **1 / 1**
Your Total Score: 31 ↑

Time Taken: **0.75**

Solve Next

[C++ STL | Set 5 \(queue\)](#) [Queue Reversal](#) [Professor and Parties](#)

```

1 // Driver Code Ends
53
54
55 /*
56 The structure of the class is
57 class MyQueue {
58 private:
59     int arr[100005];
60     int front;
61     int rear;
62
63 public :
64     MyQueue(){front=0;rear=0;}
65     void push(int);
66     int pop();
67 };
68 */
69
70 // Function to push an element x in a queue.
71 void MyQueue ::push(int x) {
72     arr[rear++]=x;
73 }
74
75 // Function to pop an element from queue and return that element.
76 int MyQueue ::pop() {
77     if(front==rear){
78         return -1;
79     }
80     int temp=arr[front];
81     front=(front+1)%100005;
82     return temp;
83 }
84
85

```

Custom Input Compile & Run Submit

Ques 3 . Implement Circular Queue using an Array

Problem link:<https://leetcode.com/problems/design-circular-queue/submissions/1577854408/>

Accepted 59 / 59 testcases passed

Vipul_04 submitted at Mar 18, 2025 19:44

Runtime 8 ms | Beats 13.02%

Memory 23.47 MB | Beats 80.08%

Code C++

```

class MyCircularQueue {
private:
    vector<int> queue;
    int front, rear, size, capacity;
public:
    MyCircularQueue(int k) {
        queue.resize(k);
        capacity = k;
    }

```

Testcase **Test Result**

Accepted Runtime: 0 ms

Case 1

Input

```

["MyCircularQueue","enqueue","enqueue","enqueue","enqueue","Rear","isFull","deQueue","enqueue","Rear"]
[[3],[1],[2],[3],[4],[],[1],[4],[1]]

```

Output

```

[null,true,true,true,false,3,true,true,true,4]

```

Expected

```

[null,true,true,true,false,3,true,true,true,4]

```

Ques 4 : Implement Two Stacks in One Array

Problem link : <https://www.geeksforgeeks.org/problems/implement-two-stacks-in-an-array/1>

The screenshot displays a coding platform interface with a problem solved successfully. The left sidebar shows the 'Output Window' with 'Compilation Results' for 'Custom Input' by 'Y.O.G.I. (AI Bot)'. The status is 'Problem Solved Successfully' with a green checkmark. Test cases passed are 1111 / 1111. Attempts are 1 / 6 with an accuracy of 16%. Points scored are 4 / 4, and the time taken is 0.04. The 'Solve Next' section lists 'Sorted subsequence of size 3', 'Move All Zeroes to End', and 'Queue using stack'. The main editor shows a C++ solution for implementing two stacks in one array. The code defines a class 'twoStacks' with methods 'push1', 'push2', 'pop1', and 'pop2' to manage two stacks using a single array.

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed
1111 / 1111

Attempts : Correct / Total
1 / 6
Accuracy : 16%

Points Scored
4 / 4
Your Total Score: 35

Time Taken
0.04

Solve Next

Sorted subsequence of size 3 Move All Zeroes to End Queue using stack

```
1 // Driver Code Ends
2
3
4
5
6
7
8
9
10
11 class twoStacks {
12 private:
13     int *arr;
14     int size;
15     int top1, top2;
16
17 public:
18     twoStacks(int n = 100) {
19         size = n;
20         arr = new int[n];
21         top1 = -1;
22         top2 = n;
23     }
24     void push1(int x) {
25         if (top1 < top2 - 1) {
26             arr[++top1] = x;
27         }
28     }
29     void push2(int x) {
30         if (top1 < top2 - 1) {
31             arr[--top2] = x;
32         }
33     }
34     int pop1() {
35         if (top1 >= 0) {
36             return arr[top1--];
37         } else {
38             return -1;
39         }
40     }
41     int pop2() {
42         if (top2 < size) {
43             return arr[top2++];
44         } else {
45             return -1;
46         }
47     }
48     ~twoStacks() {
49         delete arr;
50     }
51 }
```

Custom Input Compile & Run Submit

Ques5 : Implement queue using stack

Problem List < > 🔍

Run Submit 🛑 📄

Accepted x Editorial Solutions Submissions

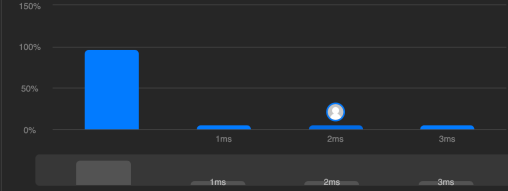
← All Submissions

Accepted 22 / 22 testcases passed
Vipul_04 submitted at Mar 18, 2025 20:04

Editorial Solution

Runtime
2 ms | Beats: 2.15%
[Analyze Complexity](#)

Memory
9.73 MB | Beats: 27.02%



Code | C++

```
class MyQueue {
private:
    std::stack<int> inputStack, outputStack;

public:
    MyQueue() {}
    void push(int x) {
        inputStack.push(x);
    }
    int pop() {
        if (outputStack.empty()) {
            while (!inputStack.empty()) {
                outputStack.push(inputStack.top());
                inputStack.pop();
            }
        }
        return outputStack.top();
    }
};
```

View more

More challenges

225. Implement Stack using Queues

Write your notes here

Code

C++ Auto

```
1 class MyQueue {
2 private:
3     std::stack<int> inputStack, outputStack;
4
5 public:
6     MyQueue() {}
7     void push(int x) {
8         inputStack.push(x);
9     }
10    int pop() {
11        if (outputStack.empty()) {
12            while (!inputStack.empty()) {
13                outputStack.push(inputStack.top());
14                inputStack.pop();
15            }
16        }
17        return outputStack.top();
18    }
19 };
```

Saved Ln 6, Col 1

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]