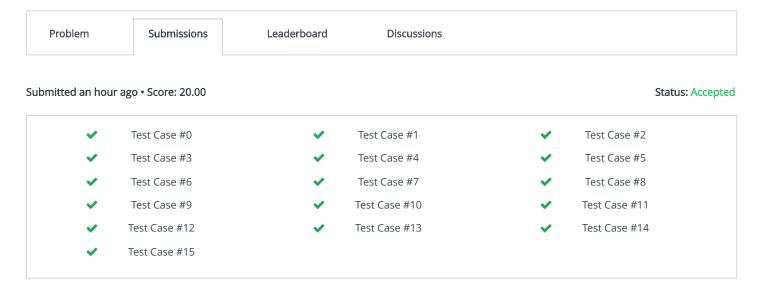
Q Search

yasinarafat2413 🗸

All Contests > Assignment 03 | Basic Data Structures | Batch 03 > Same to Same Again

Same to Same Again



Submitted Code

```
Language: C++20
                                                                                               P Open in editor
 1 #include <bits/stdc++.h>
 2 using namespace std;
 4 class Node
 5 {
 6 public:
 7
       int val;
 8
       Node *next;
9
       Node *prev;
10
11
       Node(int val)
12
13
           this->val = val;
14
           this->next = NULL;
15
           this->prev = NULL;
16
17 };
18
19 class myStack
20 {
21 public:
       Node *head = NULL;
22
23
       Node *tail = NULL;
24
       int sz = 0;
25
26
       void push(int val)
27
28
           sz++;
```

```
29
           Node *newNode = new Node(val);
30
            if (head == NULL)
31
32
            {
                head = newNode;
33
34
                tail = newNode;
35
                return;
36
           }
37
           newNode->prev = tail;
           tail->next = newNode;
38
           tail = tail->next;
39
       }
40
41
       void pop()
42
43
       {
44
            sz--;
           Node *delNode = tail;
45
46
           tail = tail->prev;
47
           if (tail == NULL)
48
            {
                head = NULL;
49
50
                return;
           }
51
52
           delete delNode;
53
           tail->next = NULL;
       }
54
55
       int Top()
56
57
       {
58
           return tail->val;
59
       }
60
61
       int size()
62
       {
63
            return sz;
64
       }
65
       bool empty()
66
67
           if (sz == 0)
68
            {
69
70
                return true;
71
72
           else
73
74
                return false;
75
       }
76 };
77
78 class myQueue
79 {
80 public:
       Node *head = NULL;
81
       Node *tail = NULL;
82
       int sz = 0;
83
84
85
       void push(int val)
86
       {
87
           sz++;
88
           Node *newNode = new Node(val);
89
           if (head == NULL)
90
91
                head = newNode;
92
                tail = newNode;
93
                return;
           }
94
```

```
95
            newNode->prev = tail;
            tail->next = newNode;
96
            tail = tail->next;
97
        }
98
99
100
        void pop()
101
        {
102
            sz--;
            Node *delNode = head;
103
            head = head->next;
104
            delete delNode;
105
            if (head == NULL)
106
107
            {
                 tail = NULL;
108
109
                 return;
            }
110
            head->prev = NULL;
111
112
        }
113
114
        int front()
115
        {
116
            return head->val;
117
        }
118
        int size()
119
        {
120
            return sz;
        }
121
122
        bool empty()
123
124
125
            if (sz == 0)
126
            {
127
                 return true;
128
            }
129
130
            else
131
                 return false;
132
        }
133 };
134
135 int main()
136 {
137
138
        int n, m;
139
        myStack st;
        myQueue q;
140
141
142
        cin >> n >> m;
143
        for (int i = 0; i < n; i++)
144
        {
145
            int x;
            cin >> x;
146
            st.push(x);
147
        }
148
149
150
        for (int j = 0; j < m; j++)
151
        {
152
            int y;
            cin >> y;
153
154
            q.push(y);
        }
155
156
157
        bool flag = true;
158
        if (n != m)
159
        {
            cout << "NO" << endl;</pre>
160
```

```
161
162
        else
163
164
        {
165
            while (!st.empty())
166
167
                 int a = st.Top();
                 int b = q.front();
168
                 if (a != b)
169
                     flag = false;
170
171
                 st.pop();
172
173
                 q.pop();
            }
174
175
             if (flag == true)
176
                 cout << "YES" << endl;</pre>
177
178
             else
179
                 cout << "NO" << endl;</pre>
        }
180
181
182
        return 0;
183 }
184
185
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |