

PREPARE^{NEW}

CERTIFY

COMPETE

Search



yasinarafat2413 ▾

[All Contests](#) > [Assignment 03](#) | [Basic Data Structures](#) | [Batch 03](#) > [Same to Same Again](#)

Same to Same Again

Problem

Submissions

Leaderboard

Discussions

Submitted an hour ago • Score: 20.00

Status: **Accepted**

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11
✓	Test Case #12	✓	Test Case #13	✓	Test Case #14
✓	Test Case #15				

Submitted Code

Language: C++20

Open in editor

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
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:
22     Node *head = NULL;
23     Node *tail = NULL;
24     int sz = 0;
25
26     void push(int val)
27     {
28         sz++;
```

```
29     Node *newNode = new Node(val);
30
31     if (head == NULL)
32     {
33         head = newNode;
34         tail = newNode;
35         return;
36     }
37     newNode->prev = tail;
38     tail->next = newNode;
39     tail = tail->next;
40 }
41
42 void pop()
43 {
44     sz--;
45     Node *delNode = tail;
46     tail = tail->prev;
47     if (tail == NULL)
48     {
49         head = NULL;
50         return;
51     }
52     delete delNode;
53     tail->next = NULL;
54 }
55
56 int Top()
57 {
58     return tail->val;
59 }
60
61 int size()
62 {
63     return sz;
64 }
65
66 bool empty()
67 {
68     if (sz == 0)
69     {
70         return true;
71     }
72
73     else
74         return false;
75 }
76 };
77
78 class myQueue
79 {
80 public:
81     Node *head = NULL;
82     Node *tail = NULL;
83     int sz = 0;
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;
96     tail->next = newNode;
97     tail = tail->next;
98 }
99
100 void pop()
101 {
102     sz--;
103     Node *delNode = head;
104     head = head->next;
105     delete delNode;
106     if (head == NULL)
107     {
108         tail = NULL;
109         return;
110     }
111     head->prev = NULL;
112 }
113
114 int front()
115 {
116     return head->val;
117 }
118 int size()
119 {
120     return sz;
121 }
122
123 bool empty()
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;
140     myQueue q;
141
142     cin >> n >> m;
143     for (int i = 0; i < n; i++)
144     {
145         int x;
146         cin >> x;
147         st.push(x);
148     }
149
150     for (int j = 0; j < m; j++)
151     {
152         int y;
153         cin >> y;
154         q.push(y);
155     }
156
157     bool flag = true;
158     if (n != m)
159     {
160         cout << "NO" << endl;
```

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