

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 class DoublyLinkedListNode {
8 public:
9     string key;
10    int value;
11    DoublyLinkedListNode *prev;
12    DoublyLinkedListNode *next;
13
14    DoublyLinkedListNode(string key, int value) {
15        this->key = key;
16        this->value = value;
17        this->prev = NULL;
18        this->next = NULL;
19    }
20
21    void removeBindings() {
22        if (this->prev != NULL) {
23            this->prev->next = this->next;
24        }
25        if (this->next != NULL) {
26            this->next->prev = this->prev;
27        }
28        this->prev = NULL;
29        this->next = NULL;
30    }
31 };
32
33 class DoublyLinkedList {
```

Solution 1 Solution 2 Solution 3

```
1 using namespace std;
2
3 // Do not edit the class below except for the insertKeyValuePair,
4 // getValueFromKey, and getMostRecentKey methods. Feel free
5 // to add new properties and methods to the class.
6 class LRUCache {
7 public:
8     int maxSize;
9
10    LRUCache(int maxSize) { this->maxSize = maxSize > 1 ? maxSize : 1;
11
12    void insertKeyValuePair(string key, int value) {
13        // Write your code here.
14    }
15
16    int *getValueFromKey(string key) {
17        // Write your code here.
18        return NULL;
19    }
20
21    string getMostRecentKey() {
22        // Write your code here.
23        return "";
24    }
25 };
26
```

Our Tests

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

Custom Output

Submit Code

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

