

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using namespace std;
4
5 class Node {
6 public:
7     int value;
8     Node *prev;
9     Node *next;
10
11     Node(int value);
12 };
13
14 class DoublyLinkedList {
15 public:
16     Node *head;
17     Node *tail;
18
19     DoublyLinkedList() {
20         head = NULL;
21         tail = NULL;
22     }
23
24     // O(1) time | O(1) space
25     void setHead(Node *node) {
26         if (head == NULL) {
27             head = node;
28             tail = node;
29             return;
30         }
31         insertBefore(head, node);
32     }
33 }
```

Solution 1 Solution 2 Solution 3

```
1 using namespace std;
2
3 class Node {
4 public:
5     int value;
6     Node *prev;
7     Node *next;
8
9     Node(int value);
10 };
11
12 // Feel free to add new properties and methods to the class.
13 class DoublyLinkedList {
14 public:
15     Node *head;
16     Node *tail;
17
18     DoublyLinkedList() {
19         head = NULL;
20         tail = NULL;
21     }
22
23     void setHead(Node *node) {
24         // Write your code here.
25     }
26
27     void setTail(Node *node) {
28         // Write your code here.
29     }
30
31     void insertBefore(Node *node, Node *nodeToInsert) {
32         // Write your code here.
33     }
34 }
```

Our Tests

Custom Output

Submit Code

```
1 def test_is_prime(n):
2     if n < 2:
3         return False
4     for i in range(2, n):
5         if n % i == 0:
6             return False
7     return True
8
9 def test_is_prime(n1, n2, n3):
10    if n1 < 2:
11        return False
12    if n2 < 2:
13        return False
14    if n3 < 2:
15        return False
16    return True
17
18 def test_is_prime(n1, n2):
19    if n1 < 2:
20        return False
21    if n2 < 2:
22        return False
23    return True
```

Run or submit code when you're ready.