

NAME: MOAZZAM FAROOQUI

ROLL NO: CT-24068

COURSE CODE: CT-159

ASSIGNMENT: DSA PRACTICE SESSION TASK#2

Q1. Winner of Circular Game

SOURCE CODE:

```
</> Code
C++ v Auto

1  class Solution {
2  public:
3      int findTheWinner(int n, int k) {
4          int winner=0;
5          for(int i=2;i<=n;i++){
6              winner=(winner+k)%i;
7          }
8          return winner+1;
9      }
10 };
...
```

ACCEPTANCE STATUS:

Testcase > Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

n =
5

k =
2

Output

3

Expected

3

Contribute a testcase

Q2. Design Browser History

SOURCE CODE:

```
</> Code
C++ Auto
1 class Node{
2     public:
3         string url;
4         Node*next;
5         Node*prev;
6         Node(string u){
7             url=u;
8             next=nullptr;
9             prev=nullptr;
10        }
11    };
12
13    class BrowserHistory {
14    private:
15        Node*current;
16    public:
17        BrowserHistory(string homepage) {
18            current=new Node(homepage);
19        }
20
21        void visit(string url) {
22            Node*newnode=new Node(url);
23            current->next=newnode;
24            newnode->prev=current;
25            current=newnode;
26        }
    }
```

```

28 ✓ string back(int steps) {
29 ✓     while(steps>0 && current->prev!=nullptr){
30         current=current->prev;
31         steps--;
32     }
33     return current->url;
34 }
35
36 ✓ string forward(int steps) {
37 ✓     while(steps>0 && current->next!=nullptr){
38         current=current->next;
39         steps--;
40     }
41     return current->url;
42 }
43 };
44
45 ✓ /**
46  * Your BrowserHistory object will be instantiated and called as such:
47  * BrowserHistory* obj = new BrowserHistory(homepage);
48  * obj->visit(url);
49  * string param_2 = obj->back(steps);
50  * string param_3 = obj->forward(steps);
51  */

```

ACCEPTANCE STATUS:

Testcase  Test Result



Accepted Runtime: 0 ms

Case 1

Input

["BrowserHistory","visit","visit","visit","back","back","forward","visit","forward","back","back"]


[["leetcode.com"],["google.com"],["facebook.com"],["youtube.com"],[1],[1],[1],["linkedin.com"],[2],[2],[7]]

Output

[null,null,null,null,"facebook.com","google.com","facebook.com",null,"linkedin.com","google.com","leetcode.com"]

Expected

[null,null,null,null,"facebook.com","google.com","facebook.com",null,"linkedin.com","google.com","leetcode.com"]

 [Contribute a testcase](#)