

```
1 #include <iostream>
2 #include <signal.h>
3 #include <string>
4 using namespace std;
5
6
7 struct DNode {
8     string name;
9     int numKills;
10    DNode* next; //person node is trying to kill
11    DNode* prev; //node someone is trying to kill
12 };
13
14 void makeCircular(DNode* list) {
15     if(list == nullptr) //can't make an empty list
16         circular
17         return;
18     DNode* temp = list;
19
20     while(temp->next != nullptr) { //links all the
21         nodes together
22         temp->next->prev = temp;
23         temp = temp->next;
24     }
25     temp->next = list; //links the head and tail
26     togther
27     list->prev = temp;
28 }
29 /**
30  *@brief just to check if makeCircular or other stuff
31  *works
32  */
33 void printList(DNode* list, int numPlayers) {
34     DNode* temp = list;
35     for(int i = 0; i < numPlayers; i++) {
36         cout << temp->name << " ";
37         temp = temp->next;
38     }
```

```

35     }
36 }
37
38 /**
39  * @brief killer is the player "previous" in the list
40  * . opposite of example where next was the killer.
41  */
42 void killPlayer(DNode* &list, string killedPlayer) {
43     DNode* temp = list;
44     while(temp->next->name != killedPlayer) { //check
45         for existence
46         if(temp->next == list) {
47             cout << killedPlayer << " does not exist
48             ." << endl << endl;
49             return;
50         }
51         temp = temp->next;
52     }
53     if(list->name == killedPlayer) {
54         list = temp;
55     }
56     DNode* deadPlayer = temp->next;
57     cout << killedPlayer << " is dead. (" <<
58     deadPlayer->numKills << " kills)" << endl;
59     temp->next = deadPlayer->next;
60     deadPlayer->next->prev = temp;
61     temp->numKills++;
62     delete deadPlayer;
63     cout << temp->name << " is the murderer. (" <<
64     temp->numKills << " kills)" << endl << endl;
65 }
66 int main()
67 {

```

```

68     cout << "Welcome to Assassin" << endl;
69     cout << "Enter the players in order (one on each
    line)." << endl;
70     cout << "When you have entered the last player (
    who is stalking the first player)," << endl;
71     cout << "enter \"QUIT\"." << endl;
72
73     DNode* head = nullptr;
74     //Enter names until player enters "QUIT"
75     //Assume no person with duplicate names
76     int numPlayers = 0;
77
78     string name;
79     getline(cin, name);
80     while(name != "QUIT") {
81         DNode* temp = new DNode();
82         //head = temp;
83         temp->numKills = 0;
84         temp->name = name;
85         temp->prev = nullptr;
86         temp->next = head;
87         head = temp;
88         getline(cin, name);
89         numPlayers++;
90     }
91
92     //Then set up circular doubly linked list
93     makeCircular(head);
94     // printList(head, numPlayers);
95
96     cout << endl << "The game begins... " << endl;
97     string killedPlayer;
98
99     while(head->next != head) { //play game in here
    until one player left
100         cout << "Who is dead? ";
101         getline(cin, killedPlayer);
102         killPlayer(head, killedPlayer);

```

```
103     }
104
105     cout << head->name << " is the Winner! (" <<
    head->numKills << " kills)" << endl;
106     cout << "Game Over";
107
108     return 0;
109 }
110
```