CircularLL... ⊗   DelAtBeg... ⊗   ⊳   ⊙ Submit

```c
1    struct node {
2        int data;
3        struct node *next;
4    };
5    typedef struct node *NODE;
6
7    NODE createNodeInCLL(){
8        NODE temp;
9        temp = (NODE) malloc(sizeof(struct node));
10       temp->next = NULL;
11       return temp;
12   }
13
14   NODE insertAtBeginInCLL(NODE first, int x)
15   {
16       NODE temp2 = createNodeInCLL();
17       temp2->data = x;
18
19       if( first == NULL )
20       {
21           temp2->next = temp2;
22           return temp2;
23       }
24
25       NODE temp = first;
26
27       while( temp->next != first )
28       {
29           temp = temp->next;
30       }
31
32       temp->next = temp2;
33
34       temp2->next = first;
35
36       return temp2;
37
38
39   }
40
41   NODE deleteAtBeginInCLL(NODE first)
42   {
43       int d;
```

⟨ Prev   Reset   Submit   Next ⟩

```c
45          if( ·first->next ·== ·first ·)
46          {
47                  printf("The ·deleted ·element ·from ·CLL ·: ·%d\n" ·, ·first->data
48          ·);
49                  return ·NULL;
50          }
51
52          NODE ·temp ·= ·first;
53
54          while( ·temp->next ·!= ·first ··)
55          {
56                  temp ·= ·temp->next;
57          }
58
59          d ·= ·first->data;
60
61          first ·= ·first->next;
62          temp->next ·= ·first;
63
64          printf("The ·deleted ·element ·from ·CLL ·: ·%d\n" ·, ·d ·);
65
66          return ·first;
67
68      }
69
70      void ·traverseListInCLL(NODE ·first) ·
71      {
72          NODE ·temp ·= ·first;
73          do ·
74          {
75                  printf("%d ·--> ·", ·temp ·-> ·data);
76                  temp ·= ·temp ·-> ·next;
77          } ·while ·(temp ·!= ·first);
78
79          printf("\n");
80      }
81
82
83
```

Terminal   Test cases