CircularLL... DelAtPos... Submit Debugger Explorer

```c
    struct node {
    int data;
    struct node *next;
};
typedef struct node *NODE;

NODE createNodeInCLL() {
    NODE temp;
    temp = (NODE) malloc(sizeof(struct node));
    temp -> next = NULL;
    return temp;
}

NODE insertAtEndInCLL(NODE first, int x)
{
    NODE temp2 = createNodeInCLL();
    temp2->data = x;

    if( first == NULL )
    {
        temp2->next = temp2;
        return temp2;
    }

    NODE temp = first;

    while( temp->next != first )
    {
        temp = temp->next;
    }

    temp->next = temp2;
    temp2->next = first;

    return first;

}

NODE deleteAtPositionInCLL(NODE first, int pos)
{
    if( pos<=0 )
    {
        printf("No such position in CLL so deletion is not
```

‹ Prev Reset Submit Next ›

```
45
46          return first;
47      }
48
49      if( pos ==1 )
50      {
51          if( first == first->next )
52          {
53              printf("The deleted element from CLL : %d\n", first-
54      >data );
55              return NULL;
56          }
57
58          printf("The deleted element from CLL : %d\n", first-
59      >data );
60
61          NODE temp = first;
62
63          while( temp->next != first )
64          {
65              temp = temp->next;
66          }
67
68          first = first->next;
69
70          temp->next = first;
71
72          return first;
73      }
74
75      NODE temp = first;
76
77      NODE temp_prev = first;
78
79      for( int i =1 ; i< (pos) ; i++ )
80      {
81          if( temp->next == first )
82          {
83              printf("No such position in CLL so deletion is not
84      possible\n");
85              return first;
86          }
87          temp_prev = temp;
88          temp = temp->next;
89      }
```

```
91      int x =temp->data;
92      temp_prev->next = temp->next;
93
94
95      printf("The deleted element from CLL : %d\n", x );
96      return first;
97  }
98
99  void traverseListInCLL(NODE first) {
100     NODE temp = first;
101     do {
102         printf("%d --> ", temp -> data);
103         temp = temp -> next;
104     } while (temp != first);
105     printf("\n");
106 }
107
108
109
```

Terminal Test cases