

Explorer

SingleLL7.c DelAtPos...

Submit

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

Plots Debugger

< Prev

Reset

Submit

Next >

```
45  NODE.deleteAtPosition(NODE.first, int.pos).
46  {
47      if(pos<=0)
48      {
49          printf("No such position in SLL so deletion is not
50  possible\n");
51          return first;
52      }
53
54      NODE.prev==first;
55      NODE.last==first;
56      if(pos==1)
57      {
58          printf("The deleted element from SLL : %d\n", first-
59  >data);
60          first==first->next;
61          return first;
62      }
63      else
64      {
65          for(int i=1; i<pos; i++)
66          {
67              if(last==NULL)
68              {
69                  printf("No such position in SLL so deletion is
70  not possible\n");
71                  return first;
72              }
73              prev==last;
74              last==last->next;
75
76          }
77
78          if(last==NULL)
79          {
80              printf("No such position in SLL so deletion is not
81  possible\n");
82              return first;
83          }
84
85          printf("The deleted element from SLL : %d\n", last-
86  >data);
87
88          prev->next==last->next;
89
90  }
```

```
91
92  v
93
94  v      return first;
95      }
96  }
97
98  void traverseList(NODE first){
99      NODE temp = first;
100      while (temp != NULL){
101          printf("%d--> ", temp->data);
102          temp = temp->next;
103      }
104      printf("NULL\n");
105  }
```



Terminal



Test cases