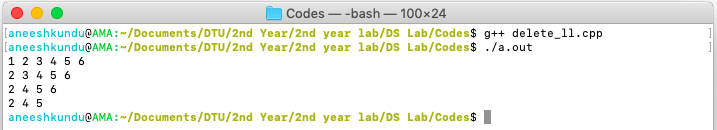
**Experiment 8**

**AIM:** Write a program to implement a linked list which deletion.

**THEORY:** A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers. If node to be deleted is root, simply delete it. To delete a node other than the head, iterate till its previous node, store address of next int temp, point its next to its next’s next, and then delete temp.

**Code:**

1. #include < iostream >
2. using namespace std;
3. struct Node {
4. int data;
5. Node \* next;
6. Node() {
7. data = 0;
8. next = NULL;
9. }
10. Node(int d) {
11. data = d;
12. next = NULL;
13. }
14. };
15. struct LinkedList {
16. Node \* head;
17. Node \* tail;
18. LinkedList() {
19. head = tail = NULL;
20. }
21. void insert(int v) {
22. //at tail
23. if (head == NULL) {
24. head = new Node(v);
25. tail = head;
26. return;
27. }
28. tail - > next = new Node(v);
29. tail = tail - > next;
30. }
31. void delete\_at(int i) {
32. if (head == NULL)
33. return;
34. Node \* cur = head, \* prev = NULL;
35. while (i--) {
36. prev = cur;
37. cur = cur - > next;
38. }
39. if (prev == NULL) {
40. Node \* temp = head - > next;
41. delete head;
42. head = temp;
43. } else {
44. if (cur - > next == NULL)
45. tail = prev;
46. prev - > next = cur - > next;
47. delete cur;
48. }
49. }
50. void print() {
51. Node \* p = head;
52. if (p == NULL)
53. return;
54. while (p != NULL) {
55. cout << p - > data << " ";
56. p = p - > next;
57. }
58. cout << endl;
59. }
60. };
61. int main() {
62. LinkedList l;
63. l.insert(1);
64. l.insert(2);
65. l.insert(3);
66. l.insert(4);
67. l.insert(5);
68. l.insert(6);
69. l.print();
70. l.delete\_at(0);
71. l.print();
72. l.delete\_at(1);
73. l.print();
74. l.delete\_at(3);
75. l.print();
76. return 0;
77. }

**Output:**