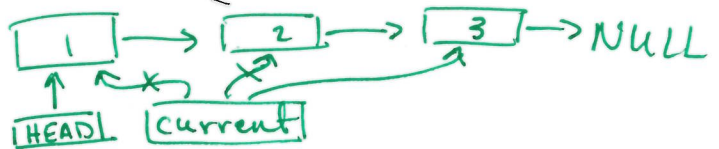
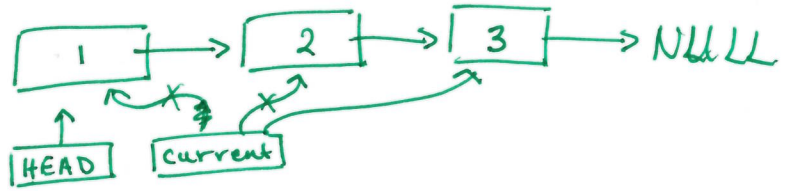
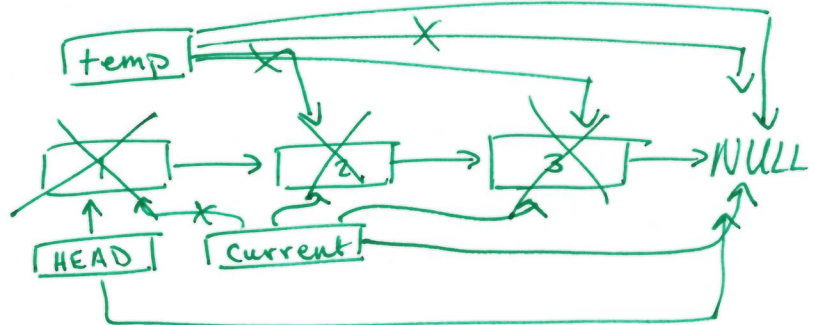


Key

```
1 #include <iostream>
2
3 using namespace std;
4
5 struct node {
6     int val;
7     node* next;
8 };
9
10 int main() {
11     //create a Linked List of 3 nodes
12     node* head = new node;
13     node* current = head;
14     for(int i=0; i<3; i++) {
15         current->val = i+1;
16         if(i == 2) {
17             current->next = NULL;
18         }
19         else {
20             current->next = new node;
21             current = current->next;
22         }
23     }
24
25     //Print the Linked List
26     current = head;
27     while(current != NULL) {
28         cout << "Node val: " << current->val << endl;
29         current = current->next;
30     }
31
32     //Delete the Linked List
33     node* temp = NULL;
34     current = head;
35     while (current != NULL) {
36         temp = current->next;
37         delete current;
38         current = temp;
39     }
40     head = NULL;
41     current = NULL;
42     return 0;
43 }
44
45
46
```



Node val: 1
Node val: 2
Node val: 3



redundant

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 void print_array(int *ar, int num) {
6     int i;
7     for(i=0; i<num; i++) {
8         printf("%d ", ar[i]);
9     }
10    printf("\n");
11 }
12
13 void get_user_input(int* ar, int num) {
14     int i;
15     for(i = 0; i < num; i++) {
16         printf("Enter a number: ");
17         scanf("%d", &ar[i]);
18     }
19 }
20
21 int main() {
22     int size = 0;
23     int* array = NULL;
24     printf("How many numbers? ");
25     scanf("%d", &size);
26     printf("%d\n", size);
27     array = (int*) malloc(sizeof(int)*size);
28     get_user_input(array, size);
29     print_array(array, size);
30     free(array);
31     return 0;
32 }
33
34
35
36
37
```

}.h only

Declarations at the top

prints a string of size
stores into the address of size
Format specifier, takes a number
what value

type cast the address

size
returns a memory address

delete

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct node {
5     int val;
6     struct node* next;
7 };
8
9 int main() {
10
11     struct node* head = (struct node *) malloc (sizeof (struct node));
12
13     struct node* current = head;
14     struct node* temp = NULL;
15     int i;
16     for(i=0; i<3; i++) {
17         current->val = i+1;
18         if(i == 2) {
19             current->next = NULL;
20         }
21         else {
22             current->next = (struct node *) malloc (sizeof (struct node));
23
24             current = current->next;
25         }
26     }
27
28     current = head;
29     while(current != NULL) {
30         printf ("val: %d ", current->val);
31
32         current = current->next;
33     }
34     printf("\n");
35
36     current = head;
37     while(current != NULL) {
38         temp = current->next;
39         free (current);
40         current = temp;
41     }
42     head = NULL;
43     current = NULL;
44
45     return 0;
46 }
47

```