Linkedlist

# Singly Linkedlist:

### 1) Delete from Beginning:

Point head to the next node i.e. second node

temp = head

head = head->next

Make sure to free unused memory

free(temp); or delete temp;

### 2) Delete from End:

Point head to the previous element i.e. last second element

Change next pointer to null

struct node \*end = head;

struct node \*prev = NULL;

while(end->next)

{

prev = end;

end = end->next;

}

prev->next = NULL;

Make sure to free unused memory

free(end); or delete end;

### 3) Delete from Middle:

Keeps track of pointer before node to delete and pointer to node to delete

temp = head;

prev = head;

for(int i = 0; i < position; i++)

{

if(i == 0 && position == 1)

head = head->next;

free(temp)

else

{

if (i == position - 1 && temp)

{

prev->next = temp->next;

free(temp);

}

else

{

prev = temp;

if(prev == NULL) // position was greater than number of nodes in the list

break;

temp = temp->next;

}

}

}