RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF EGINEERING & TECHNOLOGY

DATA STRUCTURE USING C

LECTURE -2

Umesh Kumar Gera
Assistant Professor
Computer Science & Engineering

OUTLINE

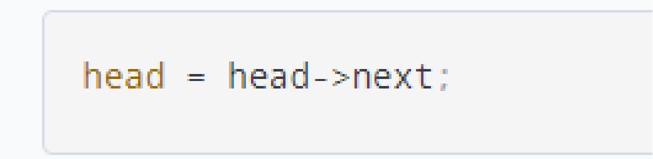
- How to Delete from a Linked List
- MCQ
- References

HOW TO DELETE FROM A LINKED LIST

How to Delete from a Linked List

Delete from beginning

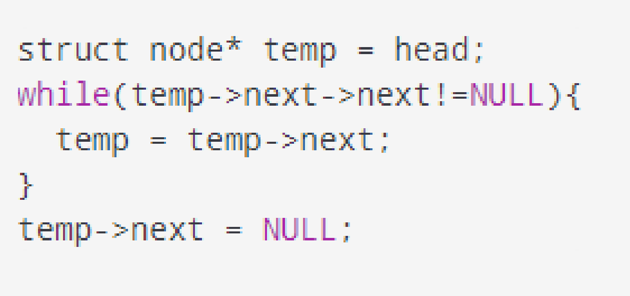
- ❑ Point head to the second node



```
head = head->next;
```

Delete from end

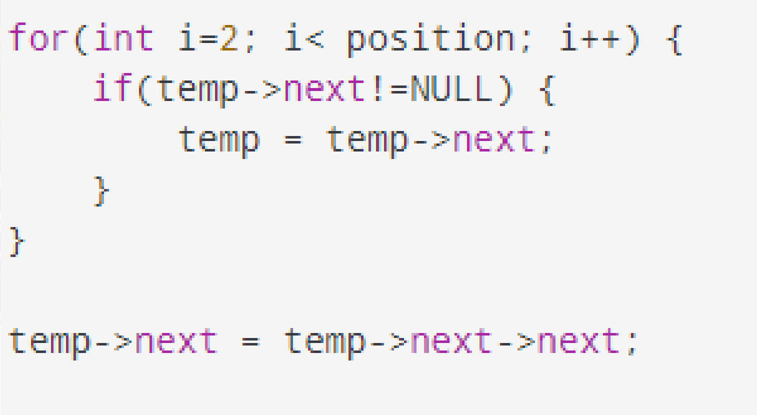
- ❑ Traverse to second last element
- ❑ Change its next pointer to null



```
struct node* temp = head;
while(temp->next->next!=NULL){
  temp = temp->next;
}
temp->next = NULL;
```

Delete from middle

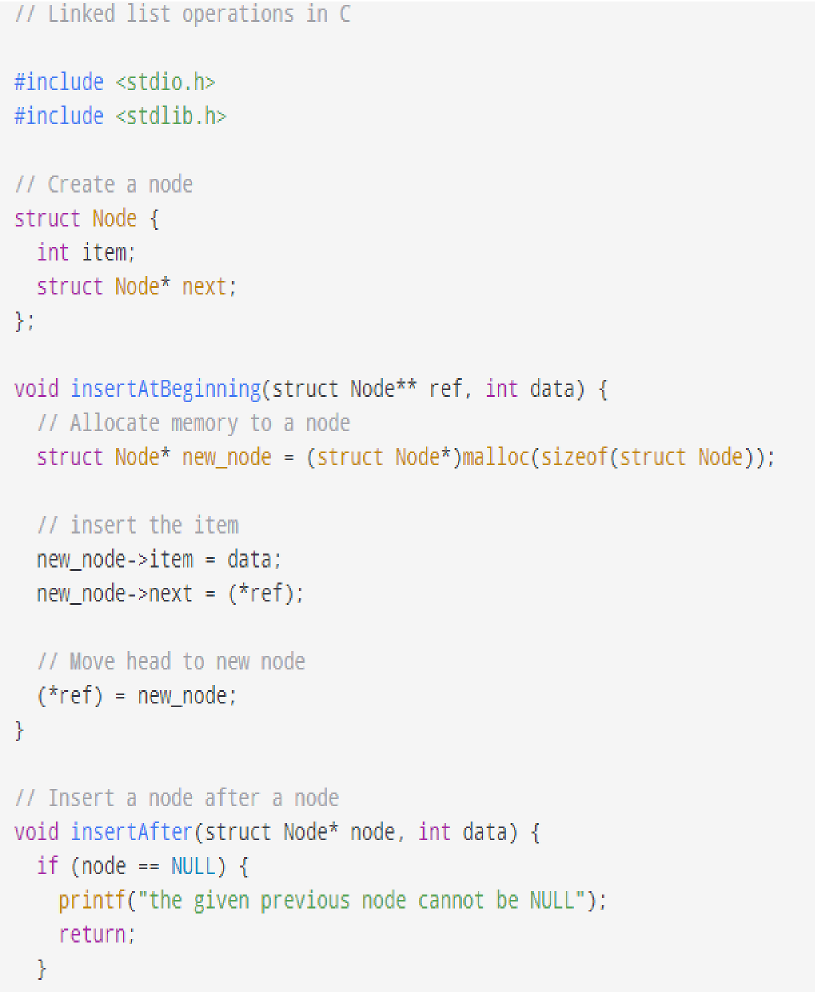
- ❑ Traverse to element before the element to be deleted
- ❑ Change next pointers to exclude the node from the chain



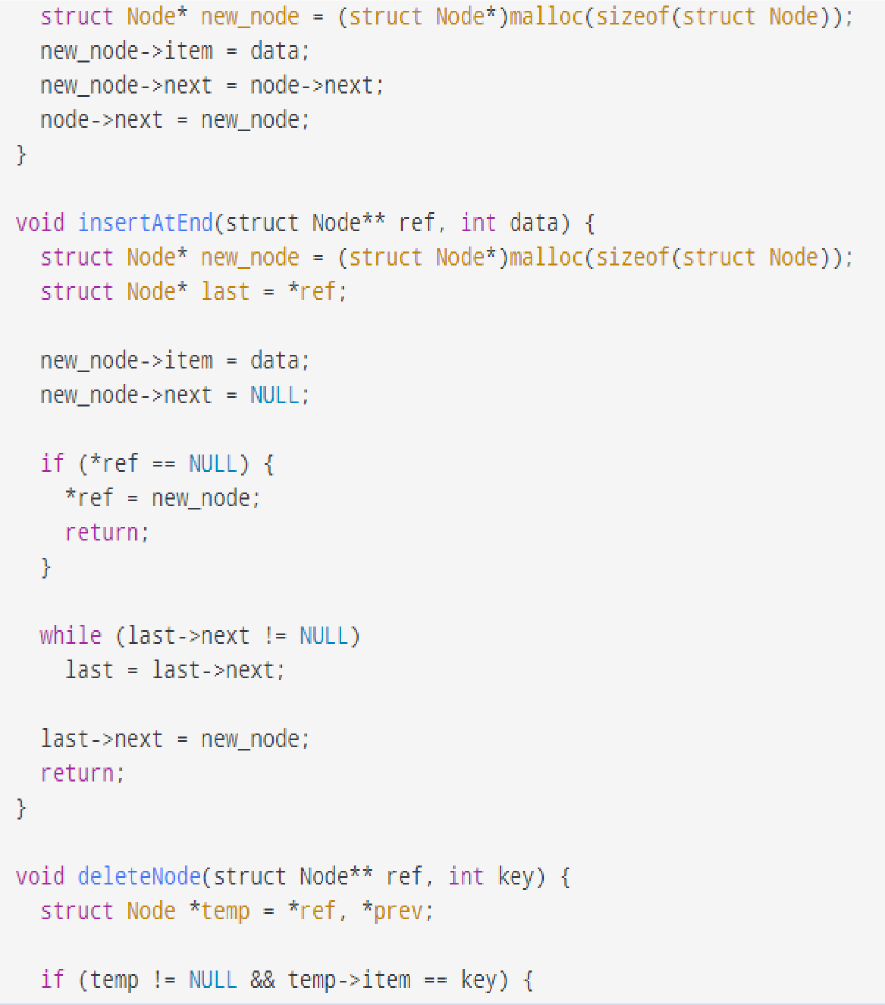
```
for(int i=2; i< position; i++) {
    if(temp->next!=NULL) {
        temp = temp->next;
    }
}

temp->next = temp->next->next;
```

HOW TO DELETE FROM A LINKED LIST

Complete Program for Linked List Operations



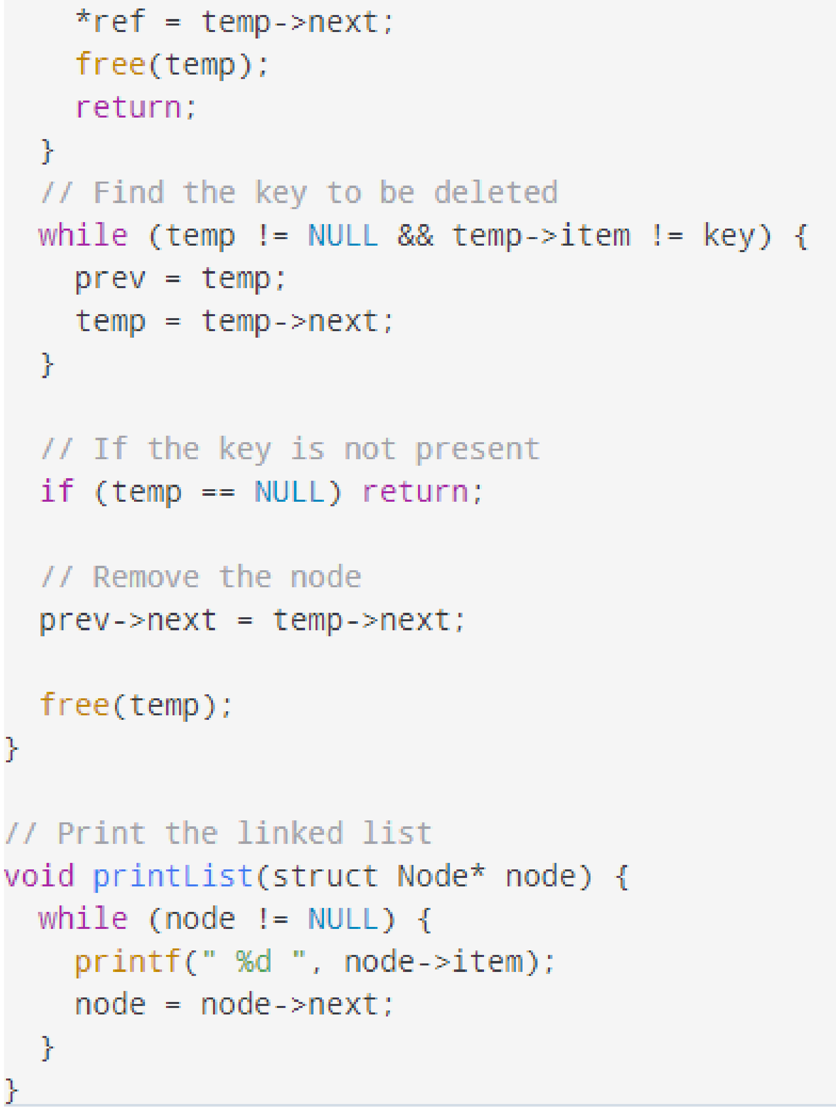
```c
// Linked list operations in C

#include <stdio.h>
#include <stdlib.h>

// Create a node
struct Node {
  int item;
  struct Node* next;
};

void insertAtBeginning(struct Node** ref, int data) {
  // Allocate memory to a node
  struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));

  // insert the item
  new_node->item = data;
  new_node->next = (*ref);

  // Move head to new node
  (*ref) = new_node;
}

// Insert a node after a node
void insertAfter(struct Node* node, int data) {
  if (node == NULL) {
    printf("the given previous node cannot be NULL");
    return;
  }
```



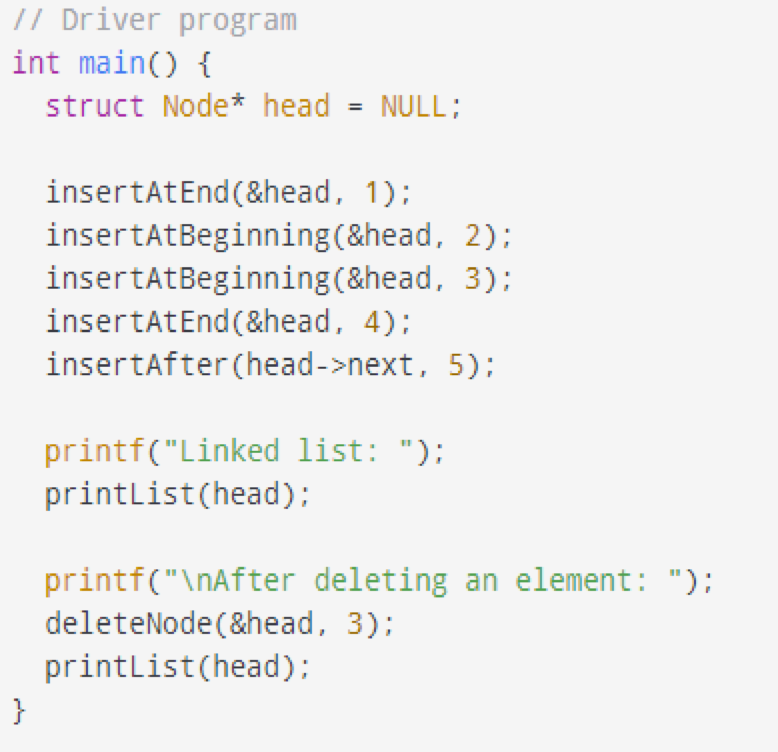
```c
  struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
  new_node->item = data;
  new_node->next = node->next;
  node->next = new_node;
}

void insertAtEnd(struct Node** ref, int data) {
  struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
  struct Node* last = *ref;

  new_node->item = data;
  new_node->next = NULL;

  if (*ref == NULL) {
    *ref = new_node;
    return;
  }

  while (last->next != NULL)
    last = last->next;

  last->next = new_node;
  return;
}

void deleteNode(struct Node** ref, int key) {
  struct Node *temp = *ref, *prev;

  if (temp != NULL && temp->item == key) {
```

HOW TO DELETE FROM A LINKED LIST

Complete Program for Linked List Operations



```
    *ref = temp->next;
    free(temp);
    return;
  }
  // Find the key to be deleted
  while (temp != NULL && temp->item != key) {
    prev = temp;
    temp = temp->next;
  }

  // If the key is not present
  if (temp == NULL) return;

  // Remove the node
  prev->next = temp->next;

  free(temp);
}

// Print the linked list
void printList(struct Node* node) {
  while (node != NULL) {
    printf(" %d ", node->item);
    node = node->next;
  }
}
```



```
// Driver program
int main() {
  struct Node* head = NULL;

  insertAtEnd(&head, 1);
  insertAtBeginning(&head, 2);
  insertAtBeginning(&head, 3);
  insertAtEnd(&head, 4);
  insertAfter(head->next, 5);

  printf("Linked list: ");
  printList(head);

  printf("\nAfter deleting an element: ");
  deleteNode(&head, 3);
  printList(head);
}
```

MCQ

QUESTIONS	OPTION A	OPTION B	OPTION C	OPTION D
This is testing question	option a testing	option b testing	option c testing	option d testing
array is a ....data structure	linear	non linear	premitive	none of these
tree is a ....data structure	linear	non linear	premitive	none of these
step by step procedure is called	methods	procedure	routine	algorithm
which of the following charecteristucs not belong to algortihm	finiteness	effectiveness	unambigous	none of these

REFERENCES

❑https://www.programiz.com/dsa/linked-list

❑https://miro.medium.com/max/3572/1*Lnb0IARMGORn_c-gYf-24g.png

❑https://www.javatpoint.com/singly-linked-list

❑https://www.tutorialspoint.com/data_structures_algorithms/linked_list_algorithms.htm