**Data Structures**

Deletion of a node in the linked list

**Deleting a node at a given position:**

* Deleting the head node(position 1).

**Head (new)Head=Head->next**

 1 add 2nd 2 add 3rd 3 NULL

* Deleting the node at any position other than position 1

 1 add 2nd 2 add 3rd 3 NULL  

* To delete the first node at position 1 make the second node as head by

Storing head->next in head.

* To delete the remaining nodes at any position other than first node traverse

Till n-1 th node(nth node is the node to be deleted) and point the next pointer

Of n-1 th node to the next pointer of nth node.

**Function for deleting a node at given position:**

//deleting node at position  
lin\_list \*DeleteNodeAtPosition(lin\_list \*head,**int** position){  
 lin\_list \*temp=head;  
 lin\_list \*temp1;  
 //linked list is empty  
 **if**(head==NULL){  
 **return** head;}  
 //deletion of first node(head node)  
 **if**(position==1){  
 head=temp->next;  
 free(temp);  
 **return** head;  
 }  
 //deletion of linked list at any position except first position  
 **for**(**int** i=1;i<position-1 && temp->next!=NULL;i++){  
 temp=temp->next;  
 }  
 **if**(temp->next==NULL ){  
 printf("invalid position\n");  
 **return** head;  
 }  
 temp1=temp->next;  
 temp->next=temp1->next;  
 free(temp1);

**return** head;}

**Whole program:**

#include<stdio.h>  
#include<stdlib.h>  
//creating a node.  
**typedef struct** lin\_list{  
 **int** data;  
 **struct** lin\_list \*next;  
}lin\_list;  
//inserting nodes  
lin\_list \*insertnode(lin\_list \*head,**int** data) {  
 lin\_list \*newnode=(lin\_list\*)malloc(**sizeof**(lin\_list));  
 newnode->data=data;  
 newnode->next=head;  
 head=newnode;  
 **return** head;  
}  
//printing the linked list.  
**void** PrintElements(lin\_list \*head){  
 //base condition  
 **if**(head==NULL){  
 **return**;  
 }  
 printf("%d ",head->data);  
 PrintElements(head->next);  
}  
//deleting node at position  
lin\_list \*DeleteNodeAtPosition(lin\_list \*head,**int** position){  
 lin\_list \*temp=head;  
 lin\_list \*temp1;  
 //linked list is empty  
 **if**(head==NULL){  
 **return** head;}  
 //deletion of first node(head node)  
 **if**(position==1){  
 head=temp->next;  
 free(temp);  
 **return** head;  
 }  
 //deletion of linked list at any position except first position  
 **for**(**int** i=1;i<position-1 && temp->next!=NULL;i++){  
 temp=temp->next;  
 }  
 **if**(temp->next==NULL ){  
 printf("invalid position\n");  
 **return** head;  
 }  
 temp1=temp->next;  
 temp->next=temp1->next;  
 free(temp1);  
 **return** head;  
}  
//main  
**int** main(){  
 lin\_list \*headA=NULL;  
 //inserting elements into linked list  
 headA=insertnode(headA,4);  
 headA=insertnode(headA,3);  
 headA=insertnode(headA,2);  
 headA=insertnode(headA,1);  
 headA=insertnode(headA,0);  
 PrintElements(headA);printf("\n");  
 //deleting node at a particular position  
 headA=DeleteNodeAtPosition(headA,3);  
 PrintElements(headA);  
 **return** 0;  
}

0 1 2 3 4

0 1 3 4