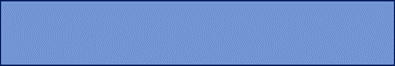
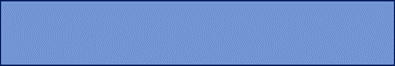
**Data Structures**

Inserting node at a particular position

**inserting at a position:**

**1 add 2nd 2 add 3rd 3 NULL**

**55 NULL**

* Create a new node.
* If linked list is empty make the new node as head.
* if position is equal to one,then point the next pointer of new node to the head

node and make the new node as head.

* In other cases where the position is greater than 1,we will traverse the

Linked list till position-1 th node and point the next of new node to

next of position-1 th node and then point the next of position-1 th node

to new node.

* Return the head(starting) node.

**Function for inserting node at position:**

lin\_list \*insertAtPosition(lin\_list \*head, **int** pos, **int** value)  
{  
 **int** i;  
 lin\_list \*temp=head, \*newnode=(lin\_list\*)malloc(**sizeof**(lin\_list));  
 newnode->data=value;  
 newnode->next=NULL;  
 **if**(head==NULL){  
 head=newnode;  
 **return** head;  
 }  
 **if**(pos==1){  
 newnode->next=head;  
 **return** newnode;  
 }  
 **for**(i=1; i<=pos-2; i++)  
 {  
 head = head->next;  
 }  
 newnode->next = head->next;  
 head->next = newnode;  
 **return** temp;  
}

**Recursive function for printing the elements of a linked list:**

* return type for the function is void.
* If linkedlist is NULL ,return nothing(void function).
* Print the data of the node.
* Recursively call the function with next node.

**void** PrintElements(lin\_list \*head){  
 **if**(head==NULL){  
 **return**;  
 }  
 printf("%d",head->data);  
 PrintElements(head->next);  
}

**Whole program:**

#include<stdio.h>  
#include<stdlib.h>  
//creating a node.  
**typedef struct** lin\_list{  
 **int** data;  
 **struct** lin\_list \*next;  
}lin\_list;  
  
lin\_list \*insertAtPosition(lin\_list \*head, **int** pos, **int** value)  
{  
 **int** i;  
 lin\_list \*temp=head, \*newnode=(lin\_list\*)malloc(**sizeof**(lin\_list));  
 newnode->data=value;  
 newnode->next=NULL;  
 **if**(head==NULL){  
 head=newnode;  
 **return** head;  
 }  
 **if**(pos==1){  
 newnode->next=head;  
 **return** newnode;  
 }  
 **for**(i=1; i<=pos-2; i++)  
 {  
 head = head->next;  
 }  
 newnode->next = head->next;  
 head->next = newnode;  
 **return** temp;  
}  
  
  
**void** PrintElements(lin\_list \*head){  
 **if**(head==NULL){  
 **return**;  
 }  
 printf("%d ",head->data);  
 PrintElements(head->next);  
}  
  
**int** main(){  
 lin\_list \*head=NULL;  
 head=insertAtPosition(head,0,1);  
 PrintElements(head);printf("\n");  
 head=insertAtPosition(head,1,2);  
 PrintElements(head);printf("\n");  
 head=insertAtPosition(head,2,3);  
 PrintElements(head);printf("\n");  
 head=insertAtPosition(head,3,55);  
 PrintElements(head);  
 **return** 0;  
}

**Output:**

1

2 1

2 3 1

2 3 55 1