1BM19CS05]
Divyanshu Thanwar

```
struct node
{
    int data;
    struct node * next;
};

struct node * head = NULL;

int length = 0;

Void insetend (int ele)
{
    struct node *newnode, *temp;
    newnode = (struct node *) malloc( sizeof(struct node));
    temp = (struct node *) malloc(sizeof (struct node));
    newnode → data = ele;
    newnode → next = NULL;
    if( head == NULL)
    {
        head = newnode;
        length = 1;
    }
    else
    {
        temp = head
        while ( temp → next != NULL)
        {
            head = head → next
            temp = temp → next;
        }
```

```
        temp->next = newnode;
        length++
    }
}

Void insertfront(int ele)
{
    struct node *temp;
    temp = (struct node*) malloc (sizeof(struct node));
    temp->data = ele;
    temp->next = head;
    head = temp;
    length++;
}

Void insertrandom(int ele, int pos)
{
    if(pos==1)
        insertfront(ele);
    else if(pos>=length+1)
        inserted(ele);
    else
    {
        struct node *temp;
        temp = (struct node*) malloc (sizeof(struct node));
        struct node *inst;
        inst = (struct node*) malloc (sizeof(struct node));
```

```
temp = head;
for(i=0; i<pos-1 ; i++)
{
        temp = temp -> next;
}
inst -> data = ele;
inst -> next = temp -> next;
temp -> next = inst;
length++;
}
}
```