

Defining The fn insert()

```

struct node
{
    int data;
    struct node *next;
};
void display(struct node *);
void insert(struct node **,int);
int main()
{
    struct node *start=NULL;
    insert(&start,10); ✓
    insert(&start,4);
    insert(&start,8);
    insert(&start,12); ✓
    ....
    display(start);
    return 0;
}

```

```

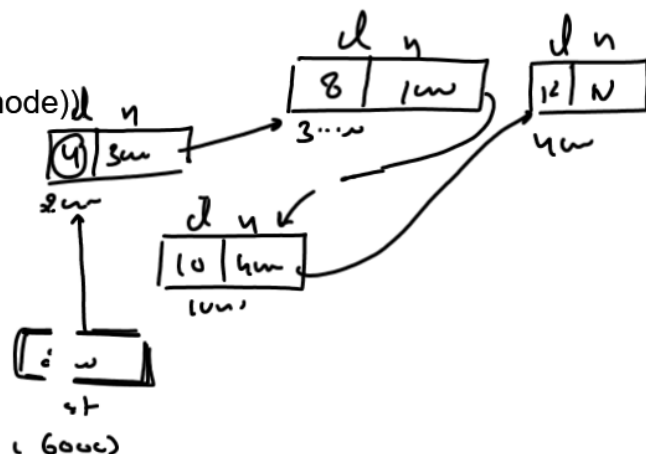
void insert(struct node **ps,int x)
{

```

```

    struct node *p,*temp,*prev;
    p=(struct node *)malloc(sizeof(struct node));
    if(p==NULL)
    {
        printf("Insufficient memory");
        return;
    }
    p->data=x;
    if(*ps==NULL)
    {
        *ps=p;
        p->next=NULL;
        return;
    }
    if(x<(*ps)->data)
    {
        p->next=*ps;
        *ps=p;
        return;
    }

```



```

temp=*ps;
while(temp!=NULL && temp->data < x)
{
    prev=temp;
    temp=temp->next;
}
if(temp==NULL)
{
    prev->next=p;
    p->next=NULL;
}
else
{
    prev->next=p;
    p->next=temp;
}
}
}

```

Writing The Function del_node()

```

void del_node(struct node **ps, int x)
{
    struct node *prev, *temp;
    if(*ps==NULL)
    {
        printf("List is empty");
        return;
    }
    temp=*ps;
    if(temp->data==x)
    {
        *ps=(temp->next);
        free(temp);
        return;
    }
}

```

```

while(temp!=NULL && temp->data!=x)
{
    prev=temp;
    temp=temp->next;
}
if(temp==NULL)
    printf("Node not found!");
else
{
    prev->next=temp->next;
    free(temp);
}
}

```

