



هياكل البيانات

المرحلة الثانية

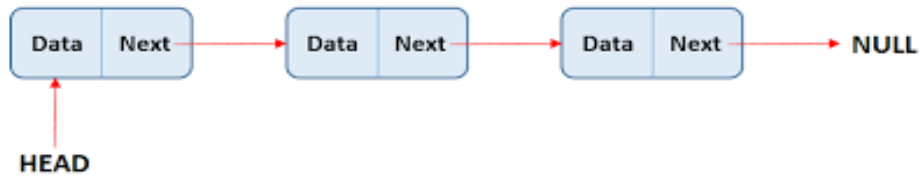
محاضرة (8)

م.م فرح معاذ جاسم

Farahmaath86@uoanbar.edu.iq

1. LINKED LIST

A Linked list is a series of connection nodes, where each node is a data structure. It can grow or shrink in size as the program runs.



Each node in a linked list contains one or more members that represent data. In addition to the data, each node contains a pointer, which can point to another node.



THE DECLARATION OF LINKED LIST

A node in a Linked List is dynamically :

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

Node *head= NULL; // this step for defined the head for any linked list .

-When we have more than one linked list, so for each linked list have a head.

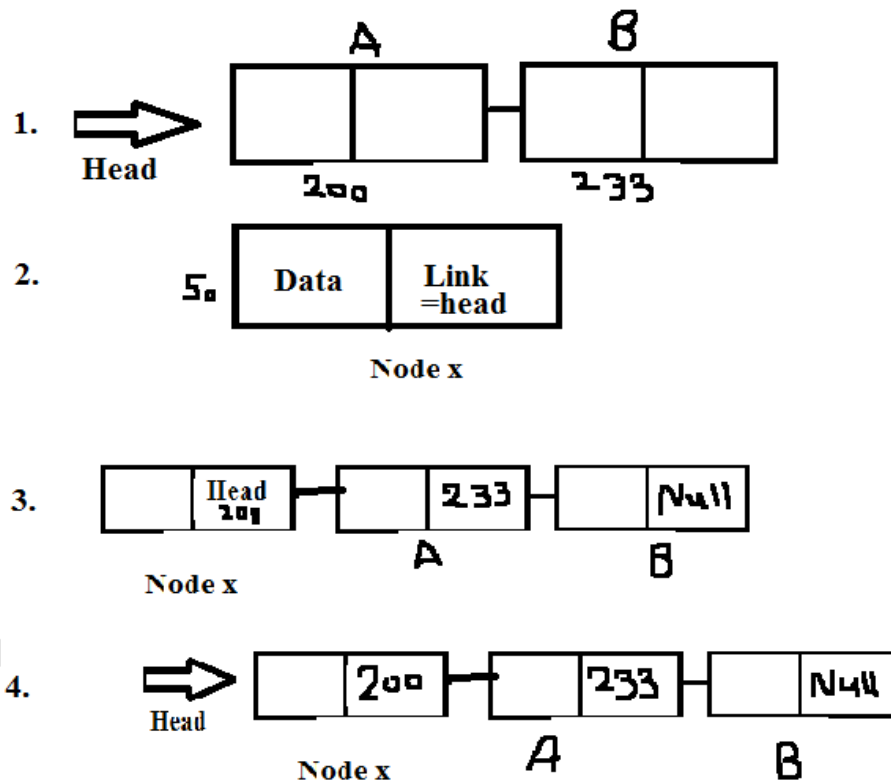
-The address of the head node give us access of the complete list.

BAICK OPRATION OF LINKED LIST

A. insert_first in a linked list need several steps

```
void insert_first(int n)
{
node *x=new node;( create new node x )
x->data= n ; (put the inserted data in node x )
x->link=head; ( make the pointer of node x take value of the head pointer)
head =x; ( node x became the first node)
```

}



B. Insert to the end of the linked list

```
void End (int n)
```

```
{
```

```
node *x=new node; //creat new node called x
```

```
x->data=n;
```

```
if (head==NULL) // لا يوجد ولا عقدة تكون العقدة الجديدة هي الاولى والاخيرة
```

```
{
```

```
head ==x;
```

```
x->link=NULL;
```

```
}
```

```
Else ( اما الحالة الثانية ننشئ عقدة جديدة بمؤشر q ثم نربطه مع السلسلة ونجعل قيمة المؤشر تساوي  
( جديد يسمى )
```

```
{ node *q=head;
```

```
While (q->link != NULL)
```

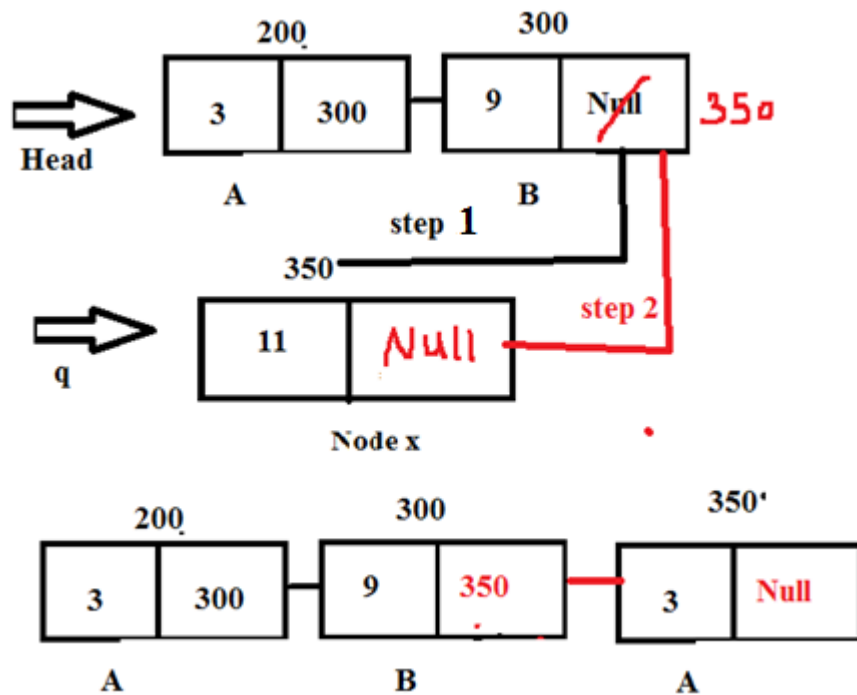
```
q=q->link;
```

```
q->link=x;
```

```
x->link=NULL;
```

```
}
```

```
}
```



B. Delete the first Node from the linked list

```
void deletF()
```

```
{
```

```
if (head->link!=NULL) ( هنا خطوة البحث عن العقدة التي تكون في البداية )
```

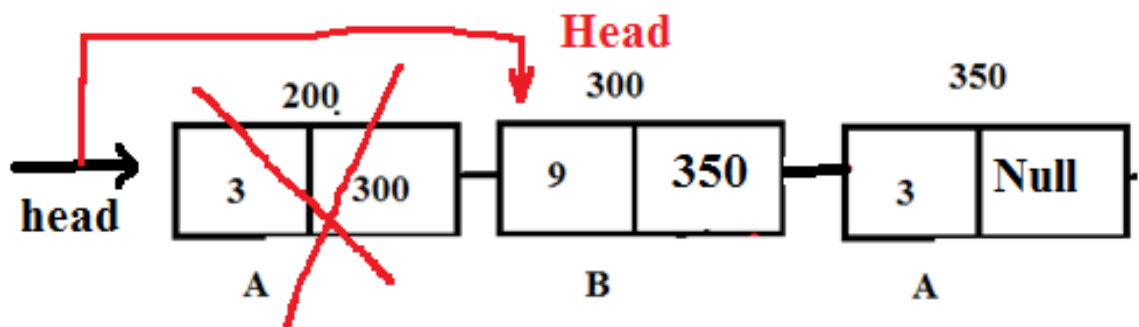
```
Head=head->link;
```

يقوم بخزن قيمة العقدة التالية للعقدة الاولى لكي لات تضيع السلسلة ويوقم بذلك تحرر العقدة الاولى ويؤشر (المؤشر على العقدة التالية)

```
else
```

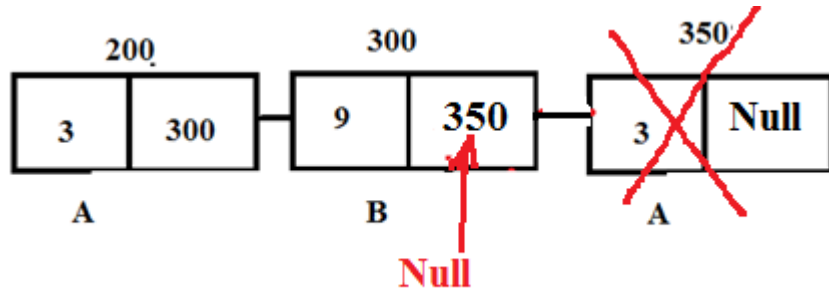
```
head=NULL;
```

```
}
```



C. Delete the end Node from the linked list

```
void DeletEnd()
{
    node *q=head;
    while (q->link->link!=NULL) (عندما يصل للنود القبل الاخيرة)
    q=q->link;
    q->link=NULL;
}
```

**D. To show the element of any linked list**

```
void display()
{
    node *q=head;

    while(q!=NULL)
    {
        cout<<q->data<<"==>";
        q=q->link;
    }
}
```