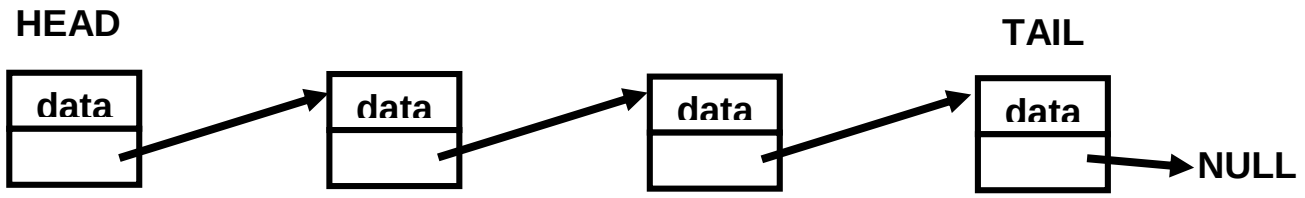
القوائم الموصولة Linked List

القوائم المتصلة هي عبارة عن هيكل بياني يتكون من مجموعة من الخلايا تسمى العقدة (node) وكل خلية تتكون من مجموع من الحقول ذات الأنواع البيانية المختلفة بالإضافة الى حقل من نوع مؤشر يستخدم لربط بين الخلايا القائمة المتصلة ويمكن توضيح القائمة المتصلة بالشكل التالي:



ويتم التصريح عن العقدة بالشكل التالي

```
struct Struct -Name
{
 Data type 1    field 1;
 Data type 2   field 2;
 node * Pointer ;
}
```

Struct : كلمة محجوز لتعريف السجلات.

Struct -Name : اسم السجل ويراعى فيه شروط تسمية المتغيرات.

field 1 , field2 : اسم الحقل ويراعى فيه شروط تسمية المتغيرات .

node * : لتصريح عن متغير من نوع مؤشر .

Pointer : اسم المؤشر ويراعى فيه شروط تسمية المتغيرات .

```
struct node
{
 char name[30];
 float avg;
 node* next;
 }
```

ملاحظات

- للوصول الى أي عقدة من عقد القائمة يجب الوصول أولا الى راس القائمة (HEAD) التي تمثل منطلقة القائمة
- في البرنامج الرئيسي يجب الإعلان عن الرأس HEAD والذيل TAIL للقائمة في البرنامج الرئيسي من نوع NODE.
- يجب إعطاء قيمة بدائية لكل من الرأس HEAD والذيل TAIL للقائمة في البرنامج الرئيسي بحيث يجب تصفيرهما

```
node* head=NULL, *tail=NULL;
```

- لتكوين خلية جديدة بمواصفات الخلية الرئيسية يجب استخدام التعليمة NEW

```
node* p=New node ;
```

- يمكن أضاف البيانات للخلية بحالتين:
  ١- عن طريق ايعاز القراءة مثل
  ٢- عن طريق الاسناد المباشر مثل

```
cin>>p->name;
p->name="Ali";
```

دوال خاصة بعمليات القوائم الموصولة هي كالتالي:

❖ برنامج فرعي لتكوين القائمة الموصولة

```
Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->no;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}
```

❖ برنامج فرعي لطباعة معلومات القائمة الموصولة

```
Void write (node* head)
{
Node* p=head;
While (p! =NULL)
{
Cout<<" The Information node: "<<endl;
Cout<< p->name;
Cout<< p->no;
P=p->link;
}}
```

❖ حذف عناصر القائمة الموصولة

```
Void del (node* &head, int n)
{
For(int i=0;i<n;i++)
{
if(head == NULL)
    cout<<"List is Empty!!!"<<endl;
  else
  {
    node* p=New node ;
    p = head;
    if(head->link == NULL)
        {
         head = NULL;
         delete(p);
        cout<<"One node deleted "<<endl;
        }
    else
        {
         head = p->link;
         delete(p);
          cout<<"One node deleted "<<endl;
        }
  }}}
```

❖ برنامج فرعي لبحث عن معلومة داخل القائمة الموصولة

```
Void search (node* &head, int key)
{
Int k=0;
 node* p1 ;
 p1 = head;
 while(p1->link != NULL)
  {
  If (p1->data ==key)
  Cout<<" This key exists in the linked list" <<endl;
   K++;
   Break;
   else
    p1 = p1 -> link;
  }
If (k==0)
  Cout<<" This key does not exist in the linked list ";
}
```

❖ برنامج فرعي لتحديث معلومة داخل القائمة الموصولة

```
Void updata (node* &head, int key)
{
Int k=0;
 node* p1 ;
 p1 = head;
 while(p1->link != NULL)
  {
  If (p1->data ==key)
  {
  Cout<<" please enter information " <<endl;
  cin>>p->name;
  k++;
  p1 = p1 -> link;
   }
  else
    p1 = p1 -> link;
  }
If (k==0)
  Cout<<" This key does not exist in the linked list ";
}
```

❖ برنامج فرعي لعمليات المعالجة

```
Void process (node* &head)
{
node* p1 ;
p1 = head;
 while(p1->link != NULL)
  {
If(p1->data >=50)
{
Cout<<"name:"<< p1->name;
Cout<< "data:"<<p1->data;
Cout<< "dep:"<<p1->dep<<endl;
p1 = p1 -> link;
}
Else
p1 = p1 -> link;
}}
```

حيث ان الشرط قابل للتغيير حسب نوع الأسئلة كان تكون حساب ناجحين او الراسبين او يقبل القسمة على مضاعفات عدد او فدي او زوجي او الاسم المختلف او الرقم الى اخره من الاختبارات

اكتب برنامج يقوم بتكوين قائمة موصولة تتضمن المعلومات التالية الاسم والقسم ثم يقوم بحذف ٣ عناصر وطباعة المتبقي؟

```
#include<iostream.h>

Struct node
   {
    Char name [35];
    Char dep [35];
    Node *link;
   };

Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}

Void del (node* &head, int n)
{
For(int i=0;i<n;i++)
{
if(head == NULL)
    cout<<"List is Empty!!!"<<endl;
  else
  {
    node* p=New node ;
    p = head;
    if(head->link == NULL)
        {
         head = NULL;
         delete(p);
```

```
        cout<<"One node deleted "<<endl;
        }
    else
        {
         head = p->link;
         delete(p);
          cout<<"One node deleted "<<endl;
         }
}}}
```

```
Void write (node* head)
{
Node* p=head;
While (p! =NULL)
{
Cout<<" The Information node: "<<endl;
Cout<< p->name;
Cout<< p->dep;
P=p->link;
}}

main()
{
node* head=NULL, *tail=NULL;
int n;
cout<<"enter value n:"<<endl;
cin>>n;
create (head,tail,n);

cout<<" Enter the number of items:"<<endl;
cin>>n;
del (head,n);

write (head);
}
```

اكتب برنامج يقوم بتكوين قائمة موصولة لمعلومات طلبه يتضمن الاسم والمعدل ثم يقوم بطباعة أسماء ومعدل الطلبة الناجحين؟

```
#include<iostream.h>
Struct node
   {
    Char name [35];
    Int avg;
    Node *link;
   };

Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->avg;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}

Void process (node* &head)
{
node* p1 ;
p1 = head;
 while(p1->link != NULL)
  {
If(p1->avg >=50)
{
Cout<<"name:"<< p1->name;
Cout<< "avg:"<<p1->avg<<endl;
p1 = p1 -> link;
}
Else
p1 = p1 -> link;
}}
```

```
main()
{
node* head=NULL, *tail=NULL;
int n;
cout<<"enter value n:"<<endl;
cin>>n;
create (head,tail,n);
process (head);
}
```

❖ حذف عناصر القائمة الموصولة القائمة الموصولة

هنالك ثلاث حالات للحذف العناصر في القائمة الموصولة وهي كالاتي :

ا- حذف عنصر في بداية القائمة الموصولة

```
void delfirst(node* &head)
{
  if(head == NULL)
    cout<<"List is Empty!!!"<<endl;
  else
  {
node* p=New node ;
    p = head;
    if(head->link == NULL)
     {
     head = NULL;
     delete(p);
    cout<<"One node deleted "<<endl;
      }
else
    {
    head = p->link;
    delete(p);
    cout<<"One node deleted "<<endl;
    }
    }
}
```

٢- حذف عنصر في نهاية القائمة الموصولة

```
void delend(node* &head)
{
  if(head == NULL)
    cout<<"List is Empty!!!"<<endl;
  else
  {
    node* p1,*p2;
    p1 = head;
    if(head->link == NULL)
    {
     head = NULL;
     delete(p1);
     delete(p2);
    }
else
    {
    while(p1->link != NULL)                \\ p1!=tail
    {
       p2 = p1;
       p1 = p1->link;
    }
     p2->link = NULL;
    delete(p1);
    cout<<"One node deleted "<<endl;
    }
}}
```

٣- حذف عنصر في القائمة الموصولة حسب معلومة

```
void delSpecific(node* &head, node*& tail ,int Value)
{
     node* p1,*p2;
    p1 = head;
  while(p1->data != Value)
  {
    p2 = p1;
    p1 = p1 -> link;
  }
if(p1 != NULL)
    {
    If(p1==head)
    Head=head->link;
Else
    If(p1==tail)
    {
    tail==p2;
    tail->link=NULL
    }
else
    p2 -> link = p1 -> link;
    delete(p1);
}
```

اكتب برنامج يقوم بتكوين قائمة موصولة تتضمن الاسم والقسم ثم يقوم بحذف اول عنصرين من القائمة وطباعة المتبقي؟

```
#include<iostream.h>

Struct node
   {
    Char name [35];
    Char dep [35];
    Node *link;
   };

Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}
void delfirst(node* &head)
{
  if(head == NULL)
    cout<<"List is Empty!!!"<<endl;
  else
  {
node* p=New node ;
    p = head;
    if(head->link == NULL)
     {
```

```
      head = NULL;
      delete(p);
     cout<<"One node deleted "<<endl;
       }
else
    {
    head = p->link;
    delete(p);
    cout<<"One node deleted "<<endl;
    }
    }
}
Void write (node* head)
{
Node* p=head;
While (p! =NULL)
{
Cout<<" The Information node: "<<endl;
Cout<< p->name;
Cout<< p->dep;
P=p->link;
}}
main()
{
node* head=NULL, *tail=NULL;
int n;
cout<<"enter value n:"<<endl;
cin>>n;
create (head,tail,n);
delfirst (head);
delfirst (head);
write (head);
}
```

❖ اضافة العناصر الى القائمة الموصولة

هنالك اربع حالات للحذف العناصر في القائمة الموصولة وهي كالاتي :

١- اضافة عنصر في بداية القائمة الموصولة

```
void addfirst(node* &head)
{
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
   p->link=head;
   head=p;
}
```

٢- اضافة عنصر في نهاية القائمة الموصولة

```
void addlast(node* &tail)
{
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    tail->link=p;
    tail=p;
   p->link=NULL;
}
```

٣- إضافة عنصر في القائمة الموصولة بعد عقدة معلومة

```
void addafter(node* &head, node* q)
{
     node* p1 ;
     p1 = head;
  while(p1->link != q->link)
  {
    p1 = p1 -> link;
  }
    node* p2=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    p2 -> link = q -> link;
    q->link=p2;
}
```

٤- إضافة عنصر في القائمة الموصولة قبل عقدة معلومة

```
void addafter(node* &head, node* q)
{
     node* p1 ;
    p1 = head;
  while(p1->link != q)
  {
   p1 = p1 -> link;
  }
    node* p2=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    p2 -> link = q;
    p1->link=p2;
}
```

اكتب برنامج يقوم بتكوين قائمة موصولة من العناصر تتضمن الاسم والقسم ثم يقوم بإضافة عنصر في بداية القائمة وعنصر في نهاية القائمة وطباعة القائمة المتكونة؟

```
#include<iostream.h>
Struct node
   {
    Char name [35];
    Char dep [35];
    Node *link;
   };
Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}
```

```
void addfirst(node* &head)
{
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
   p->link=head;
   head=p;
}
void addlast(node* &tail)
{
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
    tail->link=p;
    tail=p;
   p->link=NULL;
}
Void write (node* head)
{
Node* p=head;
While (p! =NULL)
{
Cout<<" The Information node: "<<endl;
Cout<< p->name;
Cout<< p->dep;
P=p->link;
}}
main()
{
node* head=NULL, *tail=NULL;
int n;
cout<<"enter value n:"<<endl;
cin>>n;
create (head,tail,n);
addfirst (head);
addlast (head);
write (head);
}
```

اكتب برنامج يقوم بتكوين قائمة موصولة تتضمن الاسم والقسم ثم يقوم بحذف الخلية التي تحتوي على اسم سامر وطباعة المتبقي؟

```
#include<iostream.h>

Struct node
   {
    Char name [35];
    Char dep [35];
    Node *link;
   };

Void create (node* &head, node*& tail,  int n)
{
For (int i=0; i<n; i++)
    {
    node* p=New node ;
    Cout<<"Enter the Information: "<<endl;
    cin>>p->name;
    cin>>p->dep;
    if(head==NULL)
    head=p;
    else
    tail->link=p;
    tail=p;
    tail->link= NULL
}}

    void delSpecific(node* &head, node*& tail ,char Value)
    {
      node* p1,*p2;
      p1 = head;
      while(strcmp(p1->name, Value)!=0)
      {
        p2 = p1;
        p1 = p1 -> link;
      }
    if(p1 != NULL)
       {
    If(p1==head)
    Head=head->link;
    Else
```

```
    If(p1==tail)
    {
    tail==p2;
    tail->link=NULL
    }
    else
    p2 -> link = p1 -> link;
    delete(p1);
}


Void write (node* head)
{
Node* p=head;
While (p! =NULL)
{
Cout<<" The Information node: "<<endl;
Cout<< p->name;
Cout<< p->dep;
P=p->link;
}}

main()
{
node* head=NULL, *tail=NULL;
int n;
char m="samer";
cout<<"enter value n:"<<endl;
cin>>n;
create (head,tail,n);
delSpecific (head,tail,m);
write (head);
}
```