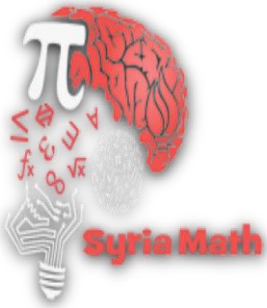


نظري

◄ دكتور المادة: سمير جعفر

◄ المحاضرة: السادسة ◄ عنوان المحاضرة: القوائم المترابطة

المحتوى العلمي : **أهلاً بكم أصدقائي سندرس في هذه المحاضرة :**

١ـ تعريف المتجهة ومثال توضيحي عنها
٢ـ كيفية بناء قاعدة مترابطة
٣ـ امثلة عما سبق

تنويه: لقد كرر الدكتور محاضرة القوائم المترابطة مرتين من اجل تأكيد وترسيخ الأفكار

- تحدثنا سابقا عن المتجهات وان اسم المتجهة عبارة عن مؤشر يؤشر على اول خانة وعناصر المتجهة تكون خانات متجاورة في الذاكرة وهذا جيد لسهولة التعامل مع الصناديق (الخانات) لكن لابد من وجود مساوئ مثلا:

نريد بناء متجهة تحجز مئة خانة في الذاكرة ولنفترض اننا اثناء تنفيذ البرنامج استخدمنا فقط 20 خانة ...

هذا سيؤدي لهدر في الذاكرة فهناك 80 خانة محجوزة دون استخدام

- من ناحية أخرى نفرض لدينا المتجهة التالية:

$int\ A[5];$ ؟؟!

يعبر عنها في الذاكرة على الشكل التالي:

∝		∝
	∝	
∝	∝	∝

لكن هذا غير صحيح لان الخانات غير متجاورة

على الرغم من انه يتوفر مكان لخمس صناديق من النوع int لكن غير متجاورة

- من المشاكل السابقة (هدر الذاكرة/التجاور) ظهرت فكرة القوائم المترابطة للدوال.

ما اهمية القوائم المترابطة؟؟

١ـ استقلال الفراغات في الذاكرة
٢ـ لا يهم التجاور في الذاكرة

Syria Math Team

٣- عند حجز متجهة نضع عدد ثابت مثلا $A[3]$ اما هنا في القوائم لا نشير الى عدد صناديق معين

٤- بناء قاعدة مترابطة عدد عناصرها مفتوح لكن بحجم الذاكرة المتاحة لدينا

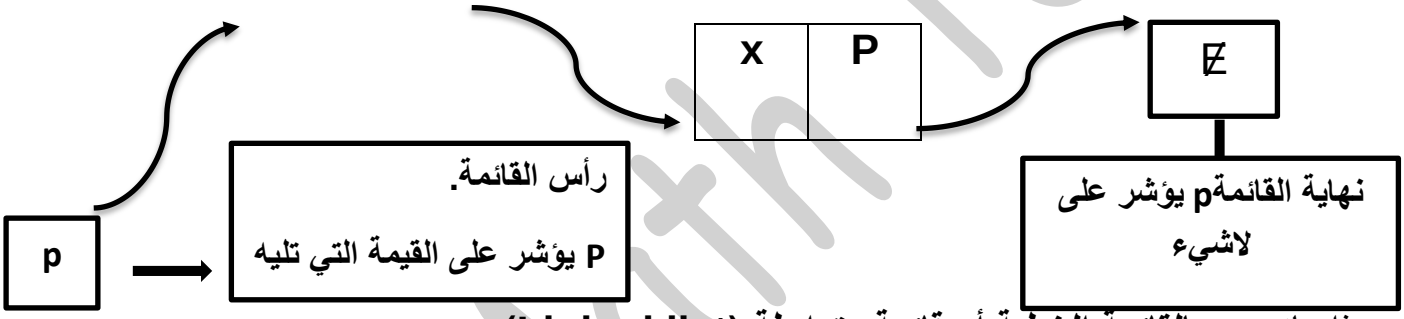
٥- نستطيع حذف العناصر سواء من اول القائمة او من اخرها

نعلم ان عنصر المتجهة يتالف من قيمة العنصر + مكان وجود العنصر التالي (مايؤشر على القيمة)

هو المؤشر

مكان وجود العنصر التالي	p	قيمة العنصر x

توضيح بسيط بالرسم :



هذا ماندعوه بالقائمة الخطية أو قائمة مترابطة (Linked list)

- لبناء قاعدة مترابطة يجب ان:

- تعريف العنصر(عناصر القائمة) (بناء النوع)
- التصريح عن مؤشر على هذا النوع (مؤشر على اول عنصرين في القائمة) مثل:
$Element * q;$
- انشاء عنصر ثم اضافته الى القائمة (حجز في الذاكرة (انشاء مكان لهذا العنصر))
$new\ Element;$ حيث ان new عنوان الخانة التي حجزت بهذه التعليمة (تحجز صندوق جديد في الذاكرة) $Element * p = new\ Element$ حيث p يؤشر على $Element$

$$p \to val = 5;$$
$$p \to next = NULL$$

$NULL$ تكتب باحرف كبيرة هي القيمة الافتراضية لمؤشر تساوي اصفر تقريبا تدل على ان المؤشر لا يؤشر على أي عنوان في الذاكرة وتكتب في نهاية القائمة لتدل على نهايتها

- ان الأسماء $Element$ (عنصر)/ $node$ (عقدة) "وهي أسماء اختيارية"

$$Element * p = new\ Element;$$

$$p \to val = 7;$$
$$p \to next = NULL;$$
$$q \to next = p;$$

❁ إضافة الى رأس القائمة (إضافة في بداية القائمة):

$$Element * p = new\ Element;$$
$$p \to val = 11;$$
$$p \to next = NULL;$$
$$p \to next = q;$$
$$q = p;$$
$$Element * s;$$
$$s = q;$$
$$while\ (s \to val! = 5)\{$$
$$s = s \to next;$$
$$\}$$

❁ بناء قاعدة خطية تحوي العناصر (2 , 5 , 3):

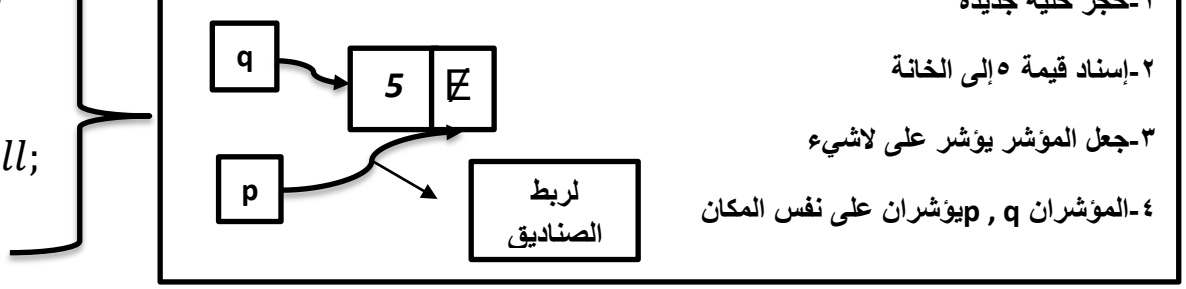
```
#include < iostream >
Using name space std;
struct node {
int val;
node * next; };
int main( ){
node * q,* p;
p = new node ;
p → val = 5;
p → next = null;
```

أنشانا سجل اسمه nodeوقيمة من النوع الصحيح (البيانات) ومؤشر له صفات السجل

مؤشرات من نوع السجل

Val : قيمة للعنصر

New:تحجز صندوق جديد في الذاكرة



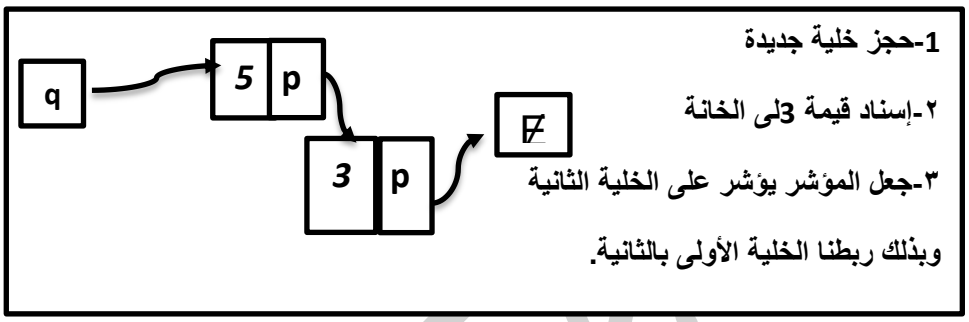
q = p;

p = new node ;

p → val = 3;

$p \rightarrow next = null;$

q → next = p;

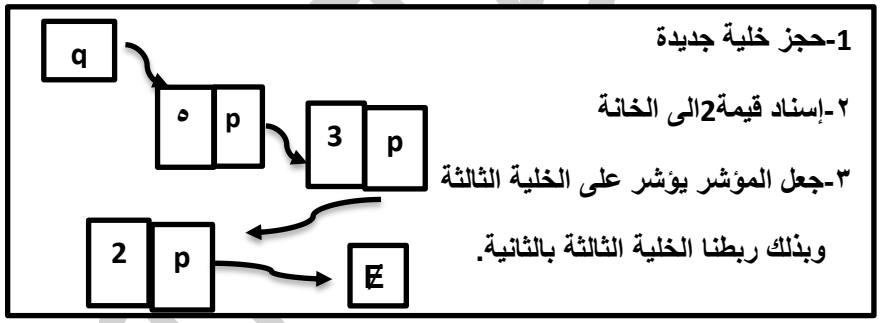


p = new node;

p → val = 2;

p → next = null;

(q → next) → $next = p;$



إذا اردنا طباعة عناصر القائمة:

node ∗ z;

z = q;

while(z = ! null)

$cout \ll z \rightarrow vall \ll endl;$

$z = z \rightarrow next;$ }

BY YOU ... and don't be a copy of the others

انتهت المحاضرة إعداد: عائشة الخطيب*علا الدلاتي*مرح غريب