**قسم الرياضيات والحاسوب / مرحلة اولى**

# هياكل برمجية Structured Programming

## المحاضرة (7)

**الهدف من المحاضرة** : هو التعرف على المصفوفات **Arrays** والصيغة العامة للأعلان عن المصفوفات، كما سيتم التعرف على طريقة فرز المعلومات باستخدام المصفوفات.

**المصفوفات**

المصفوفات هي أحد بنى البيانات الهامه، وهي عبارة عن مجموعه من خانات الذاكرة المتتالية التي لها نفس الاسم ونفس نوع البيانات. ومن أجل الرجوع إلى خانه معينه من هذه الخانات نستخدم اسم المصفوفه ورقم العنصر في المصفوفه الذي يبدأ من 0 .

**الصيغه العامة للإعلان عن المصفوفة:**

array_type array_name[elements_number];

مثلاً الإعلان int x[5]; ، يمثل إعلان عن مصفوفة من الأعداد الصحيحه بها خمسه عناصر وتتميز عناصر المصفوفه بالرقم المحصور بين قوسي المصفوفه كالآتي:

.x[0],x[1],x[2],x[3],x[4]

يمكن أن تقرأ قيم عناصر المصفوفه عن طريق لوحة المفاتيح أو تسند لها القيم عن طريق معامل الإسناد"=" كما يلي:

array_type
array_name[elements_number]={values};

يفترض أن يكون عدد العناصر مطابقا لعدد القيم وتسند القيم للعناصر على التوالي ابتداءً من العنصر الأول [0]، لكن عندما يكون عدد القيم أقل من العناصر تسند القيمة 0 تلقائياً للعناصر المتبقيه.

**ملاحظة:**

يمكن استخدام الحلقات التكراريه لتسهيل التعامل مع المصفوفات في عمليات الإدخال والإخراج أو المعالجة.

**مثال:**

اكتب برنامج لحساب معدل طالب لعشرة مواد.

```
#include<iostream.h>
  void main()
   {
      float x[10],sum=0;
      cout<<"enter the numbers";
      for(int i=0;i<10;i++)
        {
          cin>>x[i];
          sum=sum+x[i];
        }
        float   average=sum/10;
       cout<<"the summation ="<<sum;
       cout<<"\nthe average ="<<average;
       }
```

**ملاحظة:**

تستخدم المصفوفات في تمثيل سلاسل الحروف كالاسماء حيث يكون نوع المصفوفه **char**.

**المصفوفات الأحادية**

هي عبارة عن سلسلة من العناصر المتشابهة التي تخزن في الذاكرة في مواقع متجاورة والتي من الممكن الاشارة لكل واحد من هذه العناصر بشكل منفرد من خلال اضافة رقم الدلالة (index) الى الأسم التعريفي الوحيد لها، ومثلها مثل المتغيرات الاعتيادية فان المصفوفة يجب ان يتم الأعلان عنها قبل اول استخدام لها، ويكون الاعلان عن المصفوفة الاحادية بكتابة النوع اولا متبوع باسم المصفوفة كما في المتغيرات، مع اضافة قوسين مربعين بعد اسم المصفوفة يحتويان على عدد عناصر المصفوفة ( يشار له بحجم المصفوفة ايضا)، والصيغة العامة للإعلان عن المصفوفة هو:

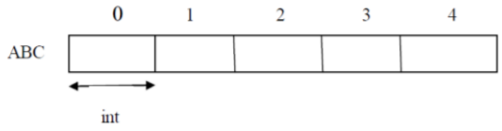
Type ArrayName [ number of elements ] ;

حيث ان النوع هو اي نوع من انواع المتغيرات المقبولة في لغة C++ ، والاسم هو اي اسم يتم اختياره من قبل المبرمج على ان يتبع القواعد المعروفة بتسمية المتغيرات، واخيرا عدد العناصر التي تحتويها المصفوفة التي يجب دائما ان يكون محددا بين قوسين مربعين، وعند الاعلان عن المصفوفة فان المترجم سيحجز عددا من المواقع المتجاورة في الذاكرة طول كل موقع ( عدد البايتات المحددة له ) يساوي الحجم المحدد لذلك النوع، وطبعا نفترض ان هذه المواقع خالية من اي قيمة او ربما هي تحتوي على قيمة قديمة ليس لها علاقة بهذا البرنامج يجب تغييرها.

**مثال:**

int ABC [ 5 ] ;

في هذا المثال سيتم تحديد خمسة مواقع متجاورة في الذاكرة من نوع الأعداد الصحيحة ( طول كل موقع 2 بايت ) خالية من القيم وتحت مسمى واحد هو (ABC) . كما في الشكل ادناه




**شكل توضيحي للمصفوفة الاحادية بعد الأعلان عنها**

لاحظ ان الأرقام التي على المصفوفة تمثل أرقام مواقع العناصر نسبة لهذه المصفوفة ودائما في مصفوفات C++ فان أول موقع يبدأ بالرقم صفر وليس واحد.

**ملاحظة:**

دائما الرقم الموجود بين القوسين المربعين والذي يمثل عدد العناصر يجب ان يكون من الاعداد الصحيحة الموجبة فقط .التعبيرات التالية غير مقبولة:

Static int value[0.02];

Float number[-90];

Char s[$];

**ملاحظة:**

قبل استخدام أي مصفوفة احادية او متعددة الابعاد في البرنامج، يجب توفير المعلومات التالية الى المترجم او المفسر

1. نوع المصفوفة ( مثلا ..... char , int, float) الخ
2. اسم المصفوفة ( يتم اختيارة من المبرمج ويفضل ان يدل على عمل المصفوفة) .
3. عدد الأبعاد subscript في المصفوفة ( هل المصفوفة احادية او متعددة الابعاد) .
4. العدد الكلي لمواقع الذاكرة المخصصة او بتحديد اكثر، عدد العناصر لكل بعد من أبعاد المصفوفة.

## انشاء المصفوفة Array Initialization

عند الاعلان عن المصفوفة فانها ستنشأ كمصفوفة خالية من القيم، حيث ان عناصرها لاتحتوي على قيم ( ربما تكون هنا قيم مخزونة من اعمال سابقة ) مالم يتم خزن قيم فيها **اي** اسناد قيم ابتدائية لهذه العناصر وتسمى هذه العملية ابتداء المصفوفة، لذلك يجب عدم أجراء اي عملية على عناصر المصفوفة اذا لم يتم اسناد قيم لها، كما هو الحال مع المتغيرات الاعتيادية.

المصفوفات ايضا يمكن ان تعرف على انها محلية او عامة مثل المتغيرات الاعتيادية، وفي كلتا الحالتين سواء كانت مصفوفة عامة او محلية، فان بامكانك اسناد قيم ابتدائية لكل عنصر من عناصرها وذلك من خلال وضع قيم بين قوسين متوسطين تفصل بين قيمة واخرى فارزة ( بنفس طريقة كتابة المجاميع ) ومساواتها الى المصفوفة كما يأتي:

```
ABC [5] = { 5, -234, 45, 0, 123 } ;
```

ويجب ان تنتبه الى ان عدد القيم بين القوسين المتوسطين يجب ان لاتزيد عن عدد عناصر المصفوفة التي تم الاعلان عنها في اعلان المصفوفة ( فاذا اعلنا عن مصفوفة من خمسة عناصر ووضعنا بين القوسين المتوسطين ستة قيم فعند ذلك سيصدر المترجم رسالة خطأ .القيم ستسند (تخزن في مواقع الذاكرة ) الى عناصر المصفوفة بالتتالي من اليسار الى اليمين ( اي ان القيمة في اقصى اليسار (5) ستسند الى العنصر في الموقع (1) ، والقيمة التي على يمينها 234- ستسند الى العنصر في الموقع (1) وهكذا.

**ملاحظة:**

عندما يتم ابتداء القيم سوف تسند لعناصر المصفوفة، C++ يسمح بامكانية الاقواس المربعه فارغة[] في هذه الحالة، فان المترجم سيفرض حجم للمصفوفة يطابق عدد القيم الموجودة بين الاقواس المتوسطة . **مثال**

```
ABC [] = { 3, 4, 5} ;
```

هنا سيحدد المترجم عدد العناصر بثلاث.

**ملاحظة:**

يجب ان تميز بين رقم موقع العنصر ومحتوى هذا الموقع، اذ ان الرقم بين القوسين المربعين هو دليل الى مكان العنصر في المصفوفة وليس لقيمة العنصر، بينما المحتوى هو يمثل القيمة التي يحتويها هذا الموقع . مثال على ذلك، لو تخيلنا شارع في منطقة سكنية يحتوي الشارع عدد من الدور السكنية ولكل دار رقم تسلسلي يمثل موقع الدار في الشارع، عليه فانه يمكنك ان تعتبر الشارع بدوره مصفوفة، فاسم الشارع يمثل اسم المصفوفة، ولكي تعنون احد الدور فتقول الدار رقم كذا في الشارع الفلاني، لتعرف ما يحتوي الدار وهي نفس طريقة الاشارة لعناصر المصفوفة ( اسم المصفوفة ثم رقم العنصر ABC [2];، لذلك فان رقم الدار لايمثل ساكني الدار فعندما تقول دار 2 فان ذلك سوف لايوضح لك عدد ساكني دار 2 وهل هم رجال او نساء او اي شيء اخر، نؤكد ان رقم الموقع هو مجرد رقم تسلسلي فقط.

**ملاحظة:**

في ادنا بعض الامثلة المقبولة لابتداء المصفوفة.

```
int value[7]={10,11,12,13,14,15,16};
float coordinate[5]={0, 0.45, -0.5, -4.0, 5.0};
char sex[2]={`M`,` F `};
char name[5]={`s`, `i`, `n`, `a`, `n`};
```

**ملاحظة:**

ان عناصر المصفوفات العامة وتلك من نوع (static) ، سوف تبدأ كليا مع القيم الافتراضية، التي هي لكل الانواع الاساسية والذي يعني املائها مع القيمة صفر. بالأمكان انشاء المصفوفة وذلك باسناد قيم لعناصر المصفوفة من لوحة المفاتيح عند تنفيذ البرنامج.

* برنامج للاعلان عن مصفوفة واسناد قيم لعناصرها من لوحة المفاتيح اثناء تنفيذ البرنامج.

```
#include<iostream>
using namespace std;
void main (void) {
int a[7];
int i ;
for ( i=0 ; i<=6 ; i++ )
cin >> a[i] ;
}
```

لاحظ اننا استخدمنا حلقة تكرار بعدد عناصر المصفوفة وذلك لكي يتم المرور على جميع مواقع المصفوفة ويسند لها قيم .اما طباعة عناصر المصفوفة فتتم بنفس الطريقة التي استخدمنا فيها حلقة التكرار لأسناد قيم لعناصر المصفوفة مع تغيير ايعاز الادخال بايعاز الاخراج .على ان تنتبه الى انه لايمكنك طباعة اي عنصر من عناصر المصفوفة اذا لم تسبقه بعملية اسناد قيم لعناصر المصفوفة.

## المصفوفات متعددة الأبعاد Multiple Subscripts Array

يمكن للمصفوفات أن تأخذ عدة أبعاد. ومن بين الاستخدامات الشائعة لهذا النوع من المصفوفات هو الجداول التي تنتظم فيها البيانات ضمن مجموعه من الصفوف و الأعمدة وبالتالي لنصل الى معلومه معينه يجب تحديد السطر والعمود وآل منهما يمثل بعداً للمصفوفه. المثال التالي يوضح الإعلان عن مصفوفه ذات بعدين:

```
float w[4][5]
```

هذه المصفوفه بها 20 عنصراً "4*5" وللتعامل مع هذه المصفوفه في عمليات الإخراج والإدخال والمعالجه نستخدم الحلقات التكراريه التي مرت معنا سابقاً.

**مثال:**

اكتب برنامجا يحسب متوسط درجات الطالب لإربعة مواد

```
            //***this program to calculate the summation****
            //***and the average of the student's degrees****
            // *************in four subjects ****************
#include<iostream.h>
   void main()
  {
  char first_name[10],last_name[10];
  int degree[4];
  float sum=0,average;
  cout<<"\n enter the first name ";
  cin>>first_name;
  cout<<"\n enter the last name ";
   cin>>last_name;
  for(int k=0;k<4;k++)
  {
   cout<<"\n enter the degree ";
   cin>>degree[k];
   sum=sum+degree[k];
   }
   average=sum/4;
   cout<<" the degree average for ";
   cout<<first_name<<" "<<last_name;
   cout<<" is = "<<average;
  }
```

نلاحظ استخدام المصفوفات في تمثيل سلاسل الحروف في الاسم الأول والثاني للطالب في البرنامج السابق.

**ملاحظة**

عند التعامل مع سلاسل الحروف ،يستخدم اسم المصفوفة الحرفية فقط .

**مثال:** اعد كتابة البرنامج السابق على أن يحسب متوسط الدرجات لعدد 6 طلاب كما في الجدول ادناه:

| اسم الطالب | المواد | | | | المتوسط |
|---|---|---|---|---|---|
| | فيزياء | رياضيات | احصاء | حاسب | |
| حسن علي | 34 | 78 | 56 | 38 | |
| علي سعد | 67 | 90 | 89 | 90 | |
| سالم علي | 45 | 23 | 78 | 76 | |
| صالح خليل | 89 | 78 | 56 | 34 | |
| محمد علي | 45 | 78 | 63 | 54 | |
| أحمد خليل | 47 | 58 | 78 | 63 | |

الجدول أعلاه يوضح درجات مجموعة طلاب في بعض المواد. لتمثيل هذه البيانات لابد لنا من التعامل مع بعدين هما الطلاب والمواد كما يظهر في البرنامج:

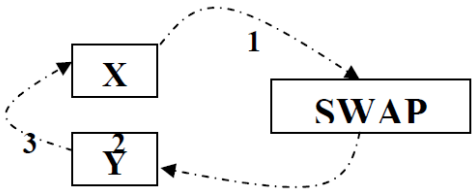
```
//***this program to calculate the summation**
//***and the average of the 6 students degrees
// *************in four subjects **************
#include<iostream.h>
  void main()
 {
 char first_name[10][6],last_name[10][6];
 int degree[4][6];
 float sum[6],average[6];
 for(int i=0;i<6;i++)
 {
 sum[i]=0;
 cout<<"\n enter the first name ";
 cin>>first_name[i];
 cout<<"\n enter the last name ";
  cin>>last_name[i];
 for(int k=0;k<4;k++)
 {
  cout<<"\n enter the degree ";
  cin>>degree[k][i];
  sum[i]=sum[i]+degree[k][i];
  }
  average[i]=sum[i]/4;
  }
  for(int j=0;j<6;j++){
  cout<<"\n the degree average for ";
  cout<<first_name[j]<<" "<<last_name[j];
  cout<<" is = "<<average[j];
  } }
```

نلاحظ استخدام مصفوفات ذات بعدين لتمثيل الدرجه *degree* و الإسم الأول *first_name* والإسم الأخير *last_name*.

## استخدام المصفوفات في فرز البيانات:

يعتبر الفرز من أهم عمليات المعالجة المستخدمة في نظم التشغيل والبرامج التطبيقية، وتوجد عدة خوارزميات للفرز، منها ما سنتناوله الآن وهو الفرز الفقاعي bubble sort والذي يعتمد على مقارنه المتغيرات فيما بينها وتبديل مواقعها بما يتناسب مع نوع الترتيب سواءً أكان تصاعدياً أم تنازلياً وهذا يتطلب مقارنة كل العناصر فيما بينها مما يستوجب استخدام حلقتين متداخلتين تعمل كل منهما n-1 مره حيث n عدد المتغيرات المراد ترتيبها.

عملية التبديل تتم باستخدام معامل الإسناد= حيث تسند القيمة الأولى لمتغير التبديل ثم القيمه الثانيه للأولى ثم قيمة متغير التبديل للثانيه كما في الرسم للتبديل بين قيمتي x,y:




يمكن كتابة الخوارزمية للترتيب التصاعدي كالآتي:

```
Loop(i=0 toi<n-1)
Loop(j=0 toj<n-1)
If(x[i]>x[i-1])
Swap=x[i]
x[i]=x[i-1]
x[i-1]=swap
```

أما الترتيب التنازلي تكون نفس الخوارزميه مع تغيير الشرط إلى:

```
If(x[i]<x[i-1])
```

**تمارين**

1. أكتب برنامج بلغة C++ يقوم بترتيب n رقم ترتيباً تصاعدياً ؟
2. أعد كتابة البرنامج 12 لعدد n طالب في m مادة بحيث يحسب متوسط درجات الطلاب في كل مادة؟