كلية العلوم

القسم : **الفيزياء**

السنة : **الثانية**

المادة : لغات البرمجة١

المحاضرة : السابعة /ن+ع/دكتور

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

المصفوفات Arrays

المصفوفات وحيدة البعد. *One dimensionsl array*

تسمى المصفوفة وحيدة البعد أو الأنساق (وأحياناً تدعى بالصف وسنتجنب هذه التسمية لتمييزها عن الصف Class)، ويعرف النسق بأنه تتابع من الأهداف كلها من النوع نفسه، هذه الأهداف تسمى عناصر النسق، ويتم ترقيمها بالتتابع من الصفر حتى n-1 لعدد n من العناصر أي 0،1،2،3،... وهذه الأرقام تسمى بالفهرس index أو القيم الجانبية للنسق .

إذا كان اسم النسق a فإن a[0] هو اسم العنصر الموجود في المكان رقم صفر (العنصر الأول)، و a[1] هو اسم العنصر الموجود في المكان رقم 1 (العنصر الثاني) وهكذا، و بصورة عامة فإن العنصر i هو العنصر الموجود في الموقع رقم i-1، وعلى ذلك فإنه إذا كان النسق يحتوي على n من العناصر فإن أسماء هذه العناصر ستكون a[0]، a[1] ، ... ، a[n-1] و يمكن تصور النسق كما في الشكل :

	0	1	2	3
a	11.11	33.33	55.55	77.77

الشكل، النسق a

يبين الشكل السابق نسقاً اسمه a مكوناً من أربعة عناصر:

العنصر الأول a[0] يحتوي على 11.11 و العنصر a[1] يحتوي على 33.33 و العنصر a[2] يحتوي على 55.55 و العنصر a[3] يحتوي على 77.77 ، وهذا الشكل ماهو في الحقيقة إلا جزء من ذاكرة الحاسب لأن أي صف يتم تخزينه عادة بهذه الطريقة بحيث تكون كل عناصره في تتابع حقيقي .

الإعلان عن النسق. Declaring array

يتم الإعلان عن النسق في لغة C++ بتحديد نوع العناصر وعددها لكي يقوم المترجم بحجز المكان الكافي ضمن الذاكرة لهذا النسق ويكون بالشكل التالي :

Type array-name [array-size] ;

حيث:

type : نوع عناصر النسق .array-size : عدد عناصر النسق .array-name : اسم النسق .

تتطلب لغة C++ القياسية أن يكون حجم النسق array-size عدداً صحيحاً موجباً

فالإعلان :

```
double a[4] ;
```

يعلن عن صف اسمه a ، عدد عناصره 4 ، نوع عناصره حقيقي .

هذا ويمكن أن يعلن عن النسق السابق بالشكل :

```
const  int size = 4 ; double  a[ size ] ;
```

حيث أعلن بداية عن size على أنه ثابت صحيح و قيمته 4 .

int a[4]، aa[5]; كما يمكن الحجز لعدة مصفوفات في نفس السطر على غرار كل النتغيرات. مثال:

إدخال عناصر النسق وإخراجها **Input and output the array elements.**

يمكن إدخال عناصر النسق و تخزينها في ذاكرة الحاسب باستخدام أي من بنى التكرار المدروسة سابقاً، وكذلك الأمر بالنسبة إلى عملية الإخراج أو الطباعة وبصورة عامة، يتم استخدام البنية التكرارية for .

يتم إدخال عناصر النسق الذي أعلن عنه سابقاً بالشكل :

```
for ( int i = 0 ; i < 4 ; i++ )
cin >> a[i] ;
```

أما الإخراج فيكون بالشكل :

```
for ( int i = 0 ; i < 4 ; i++ )
cout  << a[i] ;
```

البرنامج التالي يدخل أربعة أرقام، ثم يقوم بطباعتها بترتيب عكسي لترتيب إدخالها:

```
#include<iostream >
main(){
double a[4];
cout<<"enter 4 real number:\n";
     for(int i=1;i<=4;i++){
          cout<<i<<":";
           cin>>a[i-1];     }
cout<<"here they are in reverse order:\n";
for(i=3;i>=0;i--)
cout<<"a["<<i<<"]="<<a[i]<<endl; }
```

```
enter 4 real number:
1 : 4
2 : 5
3 : 9
4 : 1
here they are in reverse order:
a[3]=1
a[2]=9
a[1]=5
```

إعطاء قيم ابتدائية لعناصر النسق. **Initializing an array**

يمكن في لغة C++ تخصيص قيمٍ ابتدائيةٍ لأي نسق باستخدام قائمة التخصيص للقيم الابتدائية التي تحصر عناصرها بقوسي مجموعة { } ;.

مثلاً : **float a[4] = { 22.2 ، 44.4 ، 66.6 ، 88.8 } ;**

يعلن السطر السابق عن صف a عناصره من النوع الحقيقي وقد أعطيت قيماً ابتدائيةً بواسطة قائمة التخصيص للقيم الابتدائية.

مثال: البرنامج التالي يبين كيفية تخصيص قيم ابتدائية للصف.

```
#include<iostream >
void main(){
     double a[4]={ 22.2 ، 44.4 ، 66.6 ، 88.8 };
for ( int i =0 ; i<4 ; i++ )
cout<<"a["<<i<<"]="<<a[i]<<endl; }
```

```
a[0]=22.2
a[1]=44.4
a[2]=66.6
a[3]=88.8
```

نلاحظ أن قائمة القيم الابتدائية تحوي 4 عناصر وهو الحجم نفسه المحدّد في أمر إعلان النسق، لكن إذا كان للنسق عدد من العناصر أكبر من العدد الموجود في قائمة تخصيص القيم الابتدائية، فإن العناصر المتبقية يتم وضعها أصفاراً .

مثال عملي :

```
#include<iostream >
main(){
const arraySize =4;
double a[arraySize] ={ 22.2 44.4 };
for(int i=0;i< arraySize;i++)
cout<<"a["<<i<<"]="<<a[i]<<endl; }
```

```
a[0]=22.2
a[1]=44.4
a[2]=0
a[3]=0
```

مثال عملي :

```
 #include<iostream.h>
 main(){
 double a[4];
 for(int i=0;i<4;i++)
 cout<<"a["<<i<<"]="<<a[i]<<endl; }
```

```
a[0]=-9.25596e+061
a[1]=-9.25596e+061
a[2]=-9.25596e+061
a[3]=-9.25596e+061
```

خرج البرنامج:

في هذا المثال لم يتم تخصيص قيم ابتدائية لعناصر النسق وبالتالي فإن قيم عناصره تكون غير متوقعة.

وعندما يتم تخصيص قيم ابتدائية لصف فإن الإعلان عن حجمه يمكن إهماله من أمر الإعلان، فمثلا الإعلان :

```
double a[ 4 ] = { 22.2 ، 44.4 ، 66.6 ، 88.8 } ;
```

يكافئ الإعلان :

```
double a[    ] = { 22.2 ، 44.4 ، 66.6 ، 88.8 } ;
```

حيث إن حجم النسق في هذه الحالة سيحدد بعدد القيم الموجودة في قائمة تخصيص القيم الابتدائية

تطبيقات على المصفوفات الأحادية

يتم في البرنامج التالي حساب متوسط عناصر مصفوفة يتم إدخالها من لوحة المفاتيح.

مثال :

```
using namespace std;
#include <iostream >
main(){
   const int arraySize = 12;
   int a[arraySize];
   int ave; int s;
   for (int i = 0; i < arraySize ; i++){
   cin>>a[i];
   s += a[i]; }
              ave=s / arraySize;
   cout << "the average values is " << ave << endl; }
```

```
5
2
4
7
89
45
12
25
25
56
56
24
```

مثال عملي:

أوجد العنصر الأكبر في مصفوفة احادية اسمها(AA) ومكونة من 10 أرقام من النوع الحقيقي ومدخلة من لوحة المفاتيح:

```
using namespace std؛
#include <iostream<
main}()
float AA[10]; int i; float Max؛
Max=AA[0];
for (int i = 0; i < 10 ; i++)
  cin>>AA[i];
for ( i = 0; i < 10 ; i++)
if(AA[i]>Max)
Max=AA[i];
cout<<"Max="<< Max<<endl{؛
```

مثال نظري

ترتيب مصفوفة مدخلة من لوحة المفاتيح:

```
#include <iostream>
using namespace std;
int main() {
    int array[10];
    int i, j;
    int temp;
    for (i = 0; i < 10; i++)
        cin >> array[i];
    for (i = 0; i < 9; i++)           {
        for (j = i + 1; j < 10; j++) {
            if (array[j] < array[i]) {
                temp = array[i];
                array[i] = array[j];
                array[j] = temp; }}}

    cout << "The elements of array sorted:\n"
    for (i = 0; i < 10; i++)
        cout << array[i] << "\n";}
```

```
8
6
23
56
48
47
52
15
2
3
The elements of
array sorted:
2
3
6
8
15
23
47
48
52
56
```