كلية العلوم

القسم : **الفيزياء**

السنة : **الثانية**

المادة : لغات البرمجة١

المحاضرة : الثامنة /ن+ع/دكتور

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

2026

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر(What's app-Telegram)على الرقم 0931497960

المصفوفات متعددة الأبعاد Multiple subscripts arrays

كل الصفوف التي تعاملنا معها فيما سبق كانت ذات بعد واحد، ذلك يعني أنها كلها خطية أو بمعنى آخر تتابعية، ولكن في الحقيقة فإن نوع عنصر النسق يمكن أن يكون من أي نوع تقريباً بما في ذلك نوع النسق نفسه، فمثلا النسق المكون من صفوف يسمى الصف ذا الأبعاد المتعددة (مصفوفة متعددة الأبعاد) لذلك فإن صفاً ذا بعد واحد مكون من عناصر كل منها هو أيضاً صف ذا بعد واحد يمثل صفاً (مصفوفة) ذا بعدين .

كذلك فإن الصف ذي البعد الواحد المكون من عناصر كل منها صف ذو بعدين يسمى صفاً ذي ثلاثة أبعاد وهكذا.

الإعلان عن المصفوفة متعددة الأبعاد Declaring multiple subscripts arrays

يعلن عن المصفوفة متعددة الأبعاد بالشكل :

```
Type name-array [ size_Row] [ size_Col ]
```

مثال :

```
double a [ 32 ] [ 10 ] ;
```

إن هذا الإعلان يعرف مصفوفة ثنائية الأبعاد أبعادها:10 ، 32 .

والإعلان :

```
double a[ 3 ] [ 5 ] ;
```

يعرف مصفوفة ذات بعدين عدد أسطرها 3 وعدد أعمدتها 5 وسنركز في الفقرات القادمة على المصفوفة ذات البعدين.

إدخال المصفوفة ذات البعدين وإخراجها

تتم عمليتا الإدخال و الإخراج للمصفوفة ذات البعدين باستخدام حلقتي for فالمقطع البرمجي التالي يسمح بإدخال مصفوفة ذات بعدين (3×5) :

```
for ( int i = 0 ; i < 3 ; i ++ )
for ( int j = 0 ; j < 5 ; j ++ )
cin >> a[ i ] [ j ] ;
```

و المقطع البرمجي التالي يطبع مصفوفة ذات بعدين(3×5) :

```
for (i = 0 ; i < 3 ; i ++ )
for (j = 0 ; j < 5 ; j ++ )
cout << a[ i ] [ j ] ;
```

قبل كل شيء يجب كتابة المواقع للعناصر في الذاكرة كما يلي:(3*3):

0.0 0.1 0.2
1.0 1.1 1.2
2.0 2.1 2.2

1. شرط ايجاد عناصر القطر الرئيسي: `if(i==j)`
2. عناصر القر الثانوي للمصفوفة السابقة: `if(i+j=3)`
3. العناصر فوق القطر الرئيسي: `if(i<j)`
4. العناصر تحت القطر الرئيسي : `if(i>j)`
5. العناصر فوق القطر الثانوي: `if((i+j)<3)`
6. العناصر تحت القطر الثانوي: `if((i+j)>3)`
7. عناصر الصف الاول: `if(i==0)`
8. الحصول على العنصر الخامس: `if((i==1)&&(j==1))`

مثال:

اكتب برنامجا يكون خرجه الاتي:

1 1 1 1
1 10 10 1
1 10 10 1
1 1 1 1

الحل:

```
using namespace std;
#include<iostream >
void main(){
int a[3][3]; int i,j;
for(i=0;i<4;i++)
for(j=0;j<4;j++)
if((i==0)||(i==3)||(j==3)||(j==0)){
a[i][j]={1};
else
a[i][j]={10};}
for(i=0;i<4;i++){
for(j=0;j<4;j++){
cout<<a[i][j]<<"\t";}
cout<<"\n";}}
```

مثال تطبيقي عن خواص المصفوفات الثنائية (3*3) :

اكتب برنامجاً ينجز مايلي :

1- يدخل مصفوفة مؤلفة من 3 أسطر و 3 أعمدة.

2- يقوم بطباعتها سطراً سطراً.

3- طباعة عناصر القطر الرئيسي.

4- طباعة عناصر القطر الثانوي.

5- ايجاد مجموع المصفوفة؟

```
using namespace std;
#include<iostream >
main(){int a[3][3]; int i،j،s=0;
for(i=0;i<3;i++)
for(j=0;j<3;j++)
cin>>a[i][j];
for(i=0;i<3;i++)
for(j=0;j<3;j++)
cout<<a[i][j]<<" ";
cout<<endl;                                                    }
for(i=0;i<3;i++)
for(j=0;j<3;j++)
if(i==j)cout<<a[i][j]<<" ";
cout<<endl<<endl;
for(i=0;i<3;i++)
for(j=0;j<3;j++)
if(i+j==2)  cout<<a[i][j]<<" ";
cout<<endl<<endl;;
for(i=0;i<3;i++)
for(j=0;j<3;j++)
s=s+a[i][j];
cout<<"   the   sum   of   the   array   elements
="<<s<<endl; }
```

ترتيب المصفوفة الثنائية. Array sort

تعدُّ عملية فرز المعطيات (أي وضعها حسب ترتيب معين تصاعدياً أو تنازلياً) من أهم التطبيقات الحسابية، مثلاً يقوم بنك ما بفرز جميع الشيكات حسب قيمها مما يفيد إدارة البنك في استنتاج تعليمات خاصة في نهاية كل شهر. تقوم أيضاً شركات الهاتف بفرز قوائم المشتركين لديها حسب الكنية وغيرها وذلك من أجل تسهيل عملية الوصول إلى رقم هاتف شخص ما.

بصورة عامة، يجب على أية مؤسسة أن تقوم بفرز بعض المعطيات وفي بعض الحالات تكون مجبرة على فرز معطيات كبيرة لذلك تعد مسألة فرز المعطيات من المسائل المهمّة وسنقوم بدراسة نوعين للفرز هما الفرز الفقاعي والفرز الخطي.

ترتيب مصفوفة ثنائية حجمها (5*5) تصاعديا:

```
#include<iostream>
using namespace std;
int main() {
    int matrix[5][5];
                  for(int i = 0; i < 5; i++)
                  for(int j = 0; j < 5; j++)
                  cin >> matrix[i][j];
    for(int i = 0; i < 5; i++) {
    for(int j = 0; j < 5; j++) {
    for(int k = 0; k < 5; k++) {
    for(int l = 0; l < 5; l++) {

    if(matrix[k][l] < matrix[i][j]) {
                int temp = matrix[k][l];
                matrix[k][l] = matrix[i][j];
                matrix[i][j] = temp; }}}}}
    for(int i = 0; i < 5; i++) {
    for(int j = 0; j < 5; j++) {
    cout << matrix[i][j] << " ";} cout << endl;}}
```

مثال:

اكتب برنامجاً يقوم بما يلي:

1- يطبع صفاً له القيم

{ 37 ، 45 ، 89 ، 12 ، 10 ، 8 ، 2 ، 6 ، 4 }

2- يقوم بحساب مجموع أول خمسة عناصر من النسق ومتوسط تلك العناصر.

3- طباعة النسق بعد ترتيب الخمسة عناصر الأولى.

4- طباعة النسق بعد ترتيب الخمسة عناصر الأخيرة.

```
#include<iostream.h>
using namespace std;
main(){
const int n=10;
int a[n]={ 4 ،6 ،2 ،8 ،10 ،12 ،89 ،68 ،45 ،37 } ;
int hold،sum=0;float ave;
cout<<"data item in original order:"<<endl;
for(int i=0;i<n;i++)
                    cout<<a[i]<<"\t";
for(i=0;i<5;i++)
                    sum=sum+a[i];
                    ave=float(sum)/5;
cout<<"sum="<<sum<<"     "<<"  ave="<<ave;
for(int pass=0 ;pass<4;pass++)
for(i=pass+1;i<5;i++)
                   if(a[pass]>a[i]) {
                        hold=a[i];
                        a[i]=a[pass];
                        a[pass]=hold; }

for(i=0;i<n;i++)
                cout<<a[i]<<" \ t";
                cout<<endl<<endl;
```

```
for(int pass1=5 ;pass1<9;pass1++)
for(i=pass1+1;i<10;i++)
if(a[pass1]>a[i]) {

                    hold=a[i];
                    a[i]=a[pass1];
                    a[pass1]=hold; }

cout<<"data items in ascending order:"<<endl;
for(i=0;i<n;i++)
cout<<a[i] <<"\t";
cout<<endl<<endl; }
```

خرج البرنامج:

```
data item in original order:

4  6  2  8 10 12 89 68 45 37

sum=30  ave=6

data items after the arrange of the five items:

2  4  6  8 10 12 89 68 45 37

data items in ascending order:

2  4  6  8 10 12 37 45 68 89
```