المصفوفات

1- تعريف المصفوفة :

المصفوفة هي نوع من أنواع بنية البيانات، لها عدد محدود ومرتب من العناصر التي تكون جميعها من نفس النوع type، فمثلاً يمكن أن تكون جميعها أعداد صحيحة int أو حقيقية float أو محارف char ولكن لا يمكن الجمع بين نوعين مختلفين في نفس المصفوفة .

وعند تعريف المصفوفة وإنشاؤها يتم حجز عدد محدد من المواقع المتجاورة في الذاكرة لتخزين البيانات فيها ,حيث يتم الوصول الى البيانات المخزنة في هذه المواقع عن طريق اسم المصفوفة ورقم الموقع (index) والغاية من استخدام المصفوفات هي تخزين عدد من القيم تحت اسم واحد فقط (يكون لها النمط نفسه) دون الحاجة الى تخزين كل قيمة في متحول منفصل .

2- ميزات المصفوفات :

أ - تقليل حجم البرنامج .
ب - سهولة اسناد القيم واسترجاعها.
ت - استخدام تقنيات البحث والترتيب .
ث - الوصول المباشر إلى البيانات المخزنة فيها.

3- عيوب المصفوفات :

أ - لا يمكن تحديد حجمها أثناء التنفيذ(فقط يمكن تحديد حجمها عند التصريح عنها).
ب - جميع عناصرها يجب أن تكون من نوع واحد للبيانات (أي لا تحتوي على عناصر مختلفة من انواع البيانات).

4- أنواع المصفوفات :

أولاً .. المصفوفات أحادية البعد :

وتكون عبارة عن صف واحد من العناصر .
وهي تتكون من مواقع متجاورة لتخزين بيانات عددية فيها على شكل صف واحد ولكل موقع (دليل index) رقم مخصص له ويتم ترقيم هذه المواقع (الأدلة) بالتتالي (0,1,2,……,n-1) (n طول المصفوفة أي عدد عناصرها).
لتعريفها وحجز موقع لها :

```
Data Type      name of array [n];
```

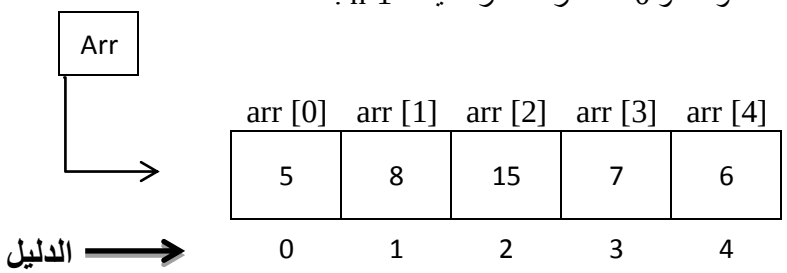
حيث :
Data Type :نوع بيانات المصفوفة .
name of array : اسم للمصفوفة .
n : عدد عناصر المصفوفة .
مثال :مصفوفة أعداد صحيحة عدد عناصرها 5 .

```
int  arr [ 5];
```

أما لإعطاء قيم ابتدائية لهذه المصفوفة :

```
arr [0]=5;
arr [1]=8;
arr [2]=15;
arr [3]=7;
arr [4]=6;
```

حيث أن دليل أول عنصر في المصفوفة هو 0 أما آخر عنصر فدليله n-1 .



أما لطباعة عنصر معين مثلاً العنصر الذي دليله 2 من المصفوفة السابقة يكون كالتالي :

```
cout<<arr[2];
```

ولتعديل القيمة المخزنة في الموقع 1 لتصبح 18 بدل من 8 نكتب :

```
arr [1]=18;
```

ولإدخال عناصر المصفوفة arr من لوحة المفاتيح :

```
cin>>arr[0];
cin>>arr[1];
cin>>arr[2];
cin>>arr[3];
cin>>arr[4];
```

وبالتالي يمكن استخدام حلقة for يبدأ العداد فيها من الـ 0 دليل أول عنصر وينتهي بأصغر تماماً من n عدد عناصر المصفوفة.

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

وكذلك الأمر من أجل طباعة عناصر مصفوفة نستخدم حلقة for :

```
for(int i=0;i<5;i++)
cout<<arr[i];
```

- أمثلة عن طريق إعطاء قيم ابتدائية لعناصر المصفوفة :
  1. يمكن إعطاء قيمة الصفر لكل عناصر المصفوفة على الشكل التالي :
     مثلاً `int A[10]={0};`
  2. يمكن تحديد القيم الابتدائية للمصفوفة أثناء التصريح عنها بالطريقة التالية :

```
int  arr [5]={5,8,15,7,6};
```

  3. يمكن إعطاء قيم ابتدائية لعناصر المصفوفة كقيمة معينة واحدة لكل العناصر بالشكل التالي :

```
# include <iostream.h>
main()
{
int A[5] ;
for (int I =0 ; I <5 ; I ++ )
A[I ] = 5 ;
return 0 ;
}
```

ملاحظات :

1. يسبب التصريح التالي :

```
int n[5] = {1 , 2 , 34 , 56 , 24 , 14} ;
```

خطأً قواعدياً لأننا أعطينا ستة قيم لمصفوفة مؤلفة من خمسة عناصر فقط .

2. يسبب التصريح التالي :

```
int n[5] = {1 , 2 , 4 , 6 , 2 } ;
```

إعطاء قيمة الصفر للعنصر الخامس من قبل المترجم .

3. إذا تم حذف حجم المصفوفة أثناء التصريح عنها فإن عدد عناصر هذه المصفوفة يصبح مساوياً لعدد القيم الابتدائية المعطاة ضمن القائمة الملحقة بالتصريح .لذلك يقوم التصريح التالي :

```
int n[ ] = { 1,2,3,4,5,6 } ;
```

بإنشاء مصفوفة مؤلفة من ستة عناصر .

مثال 1 :

اكتب برنامج لإدخال مصفوفة أعداد صحيحة عدد عناصرها 8 من لوحة المفاتيح ومن ثم طباعة هذه المصفوفة بالشكل النظامي وأيضاً طباعتها بالشكل العكسي (معكوس المصفوفة أي ابتداءً من آخر عنصر الى أول عنصر).

```
#include<iostream.h>
main()
{int a[8];
for(int i=0;i<8;i++)
  {cin>>a[i];}
for(int i=0;i<8;i++)
  {cout<<a[i]<<"\t";}
cout<<"\n";
for(int i=7;i>=0;i--)
  {cout<<a[i]<<"\t";}
return 0;}
```

ملاحظة :

تحدثنا في المحاضرة الأولى عن المتحول الثابت وهو المتحول الذي لا يمكن تغيير قيمته بعد التصريح عنه .

ويمكن استخدام المتحول الثابت في تحديد حجم المصفوفة .

```
const int size =10 ;
int s [size] ;
```

تفيد التعليمات السابقة في تحديد حجم مصفوفة s باستخدام الثابت size .

ويفيد استخدام المتحولات الثابتة لتحديد حجم المصفوفات في جعل البرامج اكثر قابلية لتغيير الحجم. فمثلاً حلقة for تقوم بتعبئة 10 عناصر يمكن تعديلها لتقوم بتعبئة 1000 عنصر وذلك بتغيير قيمة الثابت المرتبطة به أما في حالة عدم استخدام الثوابت فيتطلب التعديل السابق عدة تعديلات في اماكن مختلفة من البرنامج .

تمارين :

1. اكتب برنامج لإدخال مصفوفة أعداد صحيحة طولها 10 و طباعة الأعداد الزوجية فقط الموجودة في المصفوفة

```
# include <iostream.h>
main()
{int  num[10];
for(int i=0;i<10;i++)
{cin>>num[i];}
for(int i=0;i<10;i++)
  {  if(num[i]%2==0)
     {cout<<num[i]<<"\n";}
   }
return 0 ;}
```

2. اكتب برنامج لإدخال مصفوفة أعداد صحيحة عدد عناصرها 10 ومن ثم طباعة مجموع أعداد هذه المصفوفة .

```
# include <iostream.h>
main()
{int  num[10];
int s=0;
for(int i=0;i<10;i++)
{cin>>num[i];
s=s+num[i];}
cout<<"The Sum:"<<s;
return 0;}
```

3. اكتب برنامج جدول الضرب للعدد 4 وتخزينه في مصفوفة وطباعة قيم الجدول هذا(أي طباعة المصفوفة) .

```
# include <iostream.h>
main()
{
int  mult[11];
for(int i=0;i<11;i++)
{mult[i]=4*i;
  cout<<"4 *"<<i<<"=”<<mult[i]<< "\n";}
return 0;}
```

4. لدينا المصفوفتان الأحاديتان a[ ]={2,4,1,7} b[ ]={3,1,7,8}
اكتب برنامج لتصريح وطباعة ناتج جمع هاتين المصفوفتين

```
# include <iostream.h>
main()
{
int a[ ]={2,4,1,7};
int b[ ]={3,1,7,8};
int c[4];
   for(int i=0;i<4;i++)
       {c[i]=a[i]+b[i];
        cout<<"c["<<i<<"]="<<c[i]<<endl;}
return 0 ;
}
```

5. اكتب برنامج لتخزين مربعات الأعداد من 0 إلى 10 في مصفوفة وطباعتها .

```
# include <iostream.h>
main()
{int  sqr[11];
for(int i=0;i<=10;i++)
{sqr[i]=i*i;
  cout<<"Square "<<i<<"="<<sqr[i]<< "\n";}
return 0;}
```

6. اكتب برنامج لإدخال مصفوفة أعداد صحيحة طولها 15 ومن ثم طباعة أكبر عنصر وأصغر عنصر فيها .

```
#include <iostream.h>
 main ( )
 {   int x[15] , i, max, min ;
    for (i=0 ; i<15 ; i++)
      {cin >> x[i] ;  }
    max=x[0]; min=x[0];
     for (i=0 ; i<15 ; i++)
      {   if(x[i]>max)     {max=x[i];}
          if(x[i]<min)     {min=x[i];}
      }
cout<<”The max :”<<max <<”The min:”<<min;
return 0; }
```

7. اكتب برنامج لإنشاء مصفوفة أعداد صحيحة عدد عناصرها 7 قيم هذه المصفوفة عبارة عن العدد 1 اذا كان الدليل عدد فردي والعدد 0 اذا كان الدليل عدد زوجي .

```
#include <iostream.h>
 main ( )
 {   int num[15] ;
     for (int i=0 ; i<7 ; i++)
        {   if(i%2!=0)
                {num[i]=1;}
            else
                {num[i]=0;}
         }
         for (int i=0 ; i<7 ; i++)
            {cout<<num[i]<< "\t";}
  return 0; }
```