



الجمهورية العربية السورية

جامعة طرطوس

كلية العلوم قسم الرياضيات

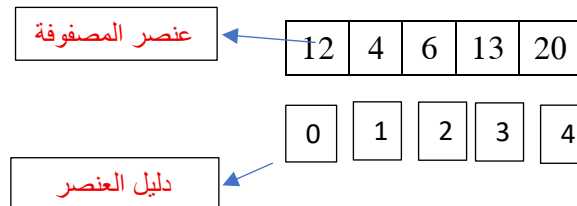
السنة الاولى

المادة: لغات البرمجة 1 – عملي

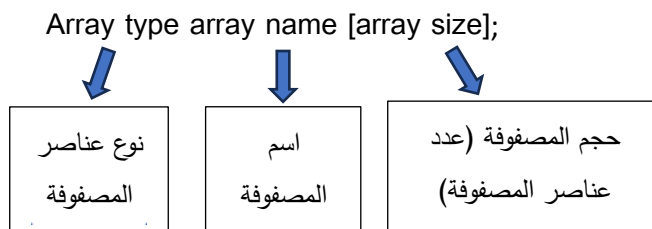
المحاضرة السابعة  
المصفوفات الأحادية

2026-2025

**المصفوفة:** هي مجموعة مواقع الذاكرة المتجاورة التي لديها اسم واحد والمخصصة لحفظ عناصر المصفوفة ، وهي من نفس النوع (ثوابت صحيحة أو حقيقية أو حرفية أو مجموعة محارف) ، ويتم الوصول إليها للكتابة فيها أو القراءة منها باستخدام دليل المصفوفة index وهو عدد صحيح موجب يحدد عنوان موقع عنصر المصفوفة وبالتالي 0، 1، 2، 3، ..... .



**التصريح عن المصفوفة:**



**مثال:**

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

أو يمكن أن نعرف متغير ونخزن ضمنه عدد عناصر المصفوفة:

```
int size = 4;
```

```
int a [size];
```

**إسناد قيم للمصفوفة:**

1. إسناد مباشر: تخزين القيم داخل المصفوفة مباشرة وقت التصريح عنها، باستخدام القوسين {}.

```
int a [5] = {12, 6, 10, 3, 8};
```

2. إدخال من لوحة المفاتيح: هي أن يدخل المستخدم القيم بنفسه أثناء تشغيل البرنامج باستخدام لوحة المفاتيح ، ويتم ذلك باستخدام الأمر cin وهنا نحتاج إلى حلقة for ونستخدمها لتكرار الإدخال لجميع عناصر المصفوفة.

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

إظهار عناصر المصفوفة: وهي أن يتم عرض العناصر الموجودة داخل المصفوفة على الشاشة ، ويتم ذلك باستخدام الأمر cout وهنا نحتاج أيضاً إلى حلقة for ونستخدمها لتكرار الطباعة لجميع عناصر المصفوفة.

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

مثال 1: أكتب برنامجاً بلغة ++C يطلب من المستخدم إدخال عشر أعداد صحيحة وتخزينها داخل المصفوفة ثم يقوم بإيجاد أكبر وأصغر قيمة ضمن عناصر المصفوفة وفي النهاية يطبع القيمتين الصغرى والكبرى على الشاشة

الحل:

```
#include<iostream.h>

int main(){
int x[10];
int max,min;
for(int i=0;i<10;i++){
cout<<"Enter element\n";
cin>>x[i];
max=x[0]; min=x[0];
for(int i=1;i<10;i++){
if(x[i]>max){
max=x[i];
}
if(x[i]<min){
min=x[i];
}
}
cout<<"max is "<<max<<endl;
cout<<"min is "<<min<<endl;
```

تعريف مصفوفة اسمها x تحوي عشر عناصر

تعريف متغيرين لتخزين القيمتين الكبرى والصغرى

حلقة for لتكرار الإدخال عشر مرات

تعيين أول عنصر كمرجع للمقارنة

حلقة مقارنة تبدأ من العنصر الثاني حتى نقارن بقية العناصر (من العنصر الثاني إلى الأخير) مع max و min ونقوم بالتحديث حسب الحاجة

طباعة القيمة الكبرى

طباعة القيمة الصغرى

}

تنفيذ البرنامج:

```
Enter element
0
Enter element
1
Enter element
2
Enter element
3
Enter element
4
Enter element
5
Enter element
6
Enter element
7
Enter element
8
Enter element
9
max is 9
min is 0
[Program finished]
```

مثال 2: اكتب برنامج يقوم بتقسيم مصفوفة أحادية يدخلها المستخدم حجمها الأعظمي 30 إلى مصفوفة أعداد زوجية ومصفوفة أعداد فردية ثم طباعة كل من المصفوفتين و مجموع عناصر كل مصفوفة وعدد عناصر كل مصفوفة.

```
#include <iostream.h>
```

```
const int s=30;
```

```
int main()
```

```
{
```

```
int n, k=0, l=0;
```

```
int sum1=0, sum2=0;
```

```
int a[s], b[s], c[s];
```

```
cout<<"enter array size";
```

```
cin>>n;
```

```
for (int i=0; i<n;i++){
```

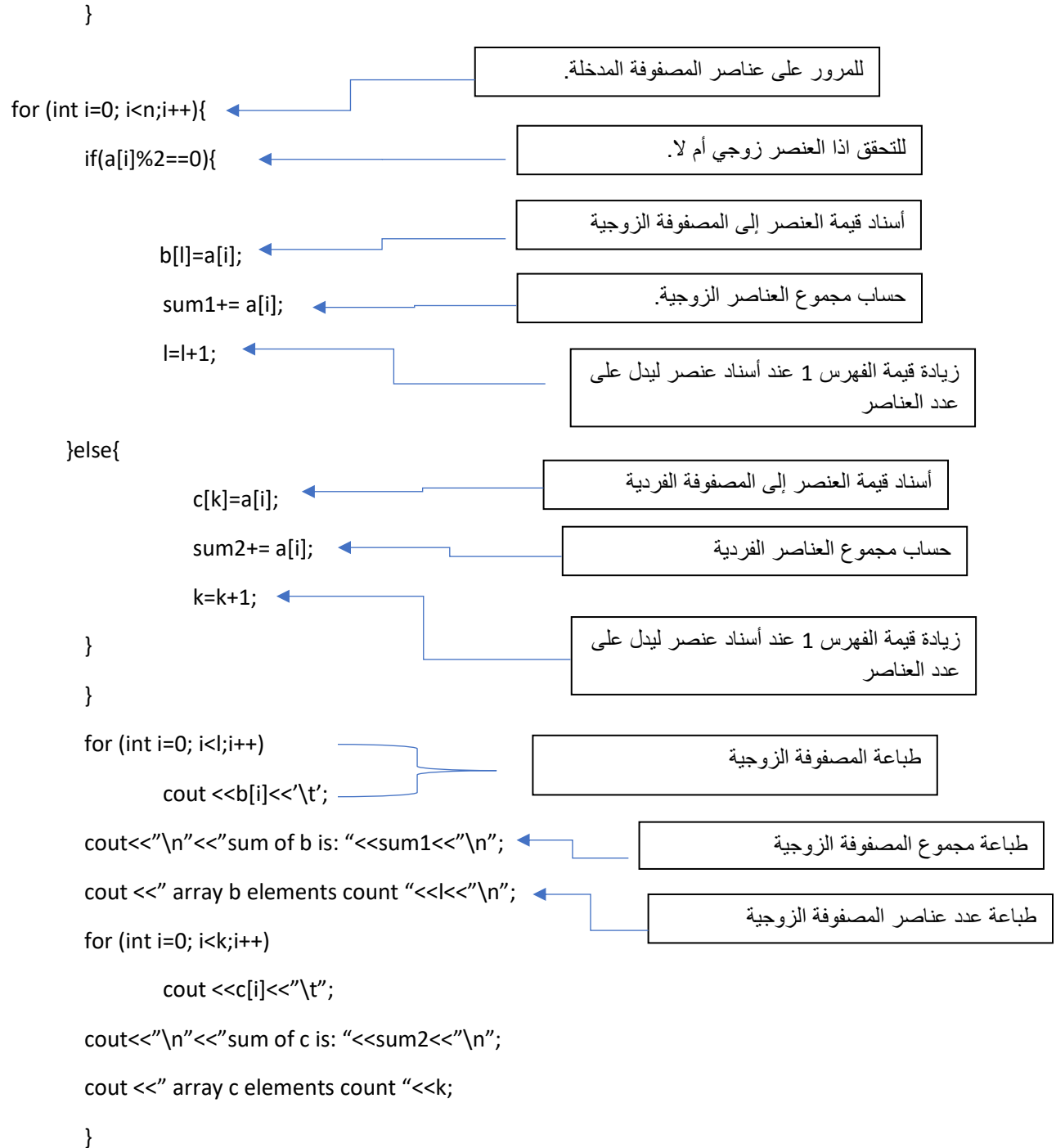
```
cout<<"enter array element";
```

```
cin>>a[i];
```

تعريف متغير لتحديد حجم المصفوفة الأعظمي

تعريف المتغيرات لتحديد حجم كل مصفوفة. يمثل k فهرس المصفوفة الفردية وتم اسناد قيمة ابتدائية له تساوي الصفر لأن عند عملية إسناد قيم للمصفوفة سنبدأ بالفهرس 0. l هو فهرس المصفوفة الزوجية.

تم تهيئة المصفوفات بالحجم الافتراضي. b المصفوفة الزوجية، c المصفوفة الفردية.



تنفيذ البرنامج:

```

enter array size6
enter array element7
enter array element4
enter array element8
enter array element2
enter array element1
enter array element1
4      8      2
sum of b is: 14
array b elements count 3
7      1      1
sum of c is: 9
array c elements count 3
[Program finished]

```

مثال 3: اكتب برنامج يقوم باختبار إذا المصفوفة متناظرة أم لا.

نقوم بمقارنة العنصر الأول مع العنصر الأخير، والعنصر الثاني مع العنصر قبل الأخير وهكذا، بما أننا نقارن العنصر مع العنصر المقابل له يكفي المرور على نصف العناصر المصفوفة

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
int f=0;
```

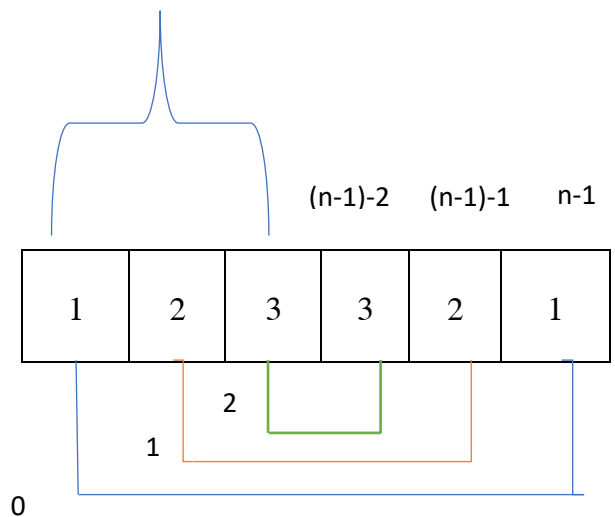
```
int n=6;
```

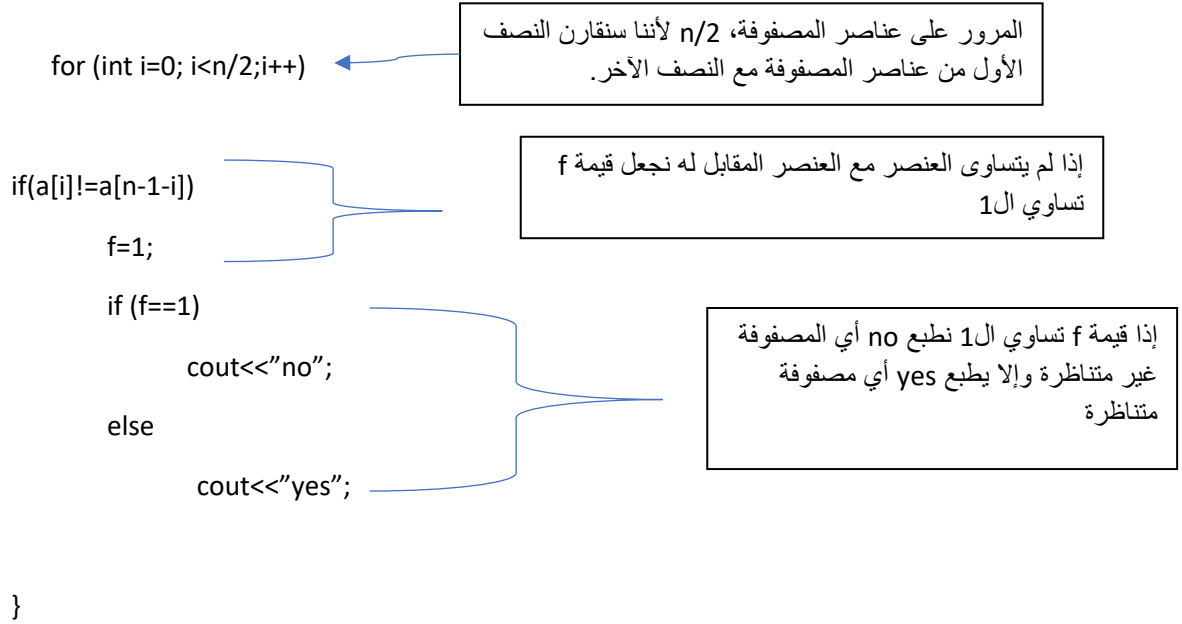
```
int a[n];
```

```
for (int i=0; i<n;i++)
```

```
cin>>a[i];
```

متغير لإختبار  
تناظر المصفوفة





تنفيذ البرنامج:

```
8
9
8
5
7
3
no
[Program finished]
```

```
7
3
2
2
3
7
yes
[Program finished]
```