جَامعة المَنارة
MANARA UNIVERSITY

المعلوماتية كلية الهندسة

مقرر مدخل إلى الخوارزميات والبرمجة
Introduction to Algorithms and Programming

إ. د. علي عمران سليمان

محاضرات الأسبوع التاسع

الفصل الثاني 2022-2023

المصفوفات Arrays

عناوين الفصل :

4-0- مقدمة.	4-1-4-3 توليد الأعداد العشوائية.
4-1- المصفوفات وحيدة البعد one. dimension array	4-1-4-4- لعبة النرد Dice-rolling
4-1-1-الإعلان عن النسقDeclaring Array	4-2- المصفوفات متعددة الأبعاد Multiple dimension arrays
4-1-2- إدخال و إخراج عناصر النسق.	4-2-1- الإعلان عن المصفوفة متعددة الأبعاد Declaring multiple dimension arrays
4-1-3- إعطاء قيم ابتدائية لعناصر النسق.	4-2-2- إدخال المصفوفة وإخراجها
4-1-4- تطبيقات على المصفوفات.	4-3- فرز النسق
4-1-4-1-متوسط نسق	4-4- البحث ضمن النسق
4-1-4-2-المخطط البياني	

المحاضرة من المراجع :

- Deitel & Deitel, C++ How to Program, Pearson; 10th Edition (February 29, 2016)
- **د.علي سليمان, مدخل إلى الحاسوب والخوارزميات, جامعة تشرين 2006-2005**

4-0- مقدمة introduction

المتحول Var : يخزن قيمة واحدة ومن نوع واحد فقط،
المصفوفة Array : تخزن مجموعة من قيم العناصر تحت اسم واحد، وتعرف بأنها بنية معطيات تجَمِّيعه من نوع واحد.
data structure aggregated from one type.
السجلات struct والاصناف classes التي يمكن أن تملك ضمن بنيتها عدة أنواع من المعطيات والبرمجيات (تدرس لاحقاً).
ومن مساوئ المصفوفات والسجلات والاصناف أنها تملك حجوم ثابتة طيلة فترة تنفيذ البرنامج، إلا إذا كان الحجز من النمط الديناميكي، والتي تخلق وتدمر عند الدخول إلى كتلتها والخروج منها على الترتيب.

تحجز المصفوفة منطقة في الذاكرة تتكون من عدد محدود Limited ومتجانس Homogeneous من المواقع المستمرة continuous
أ – محدود: رقم صحيح موجب ثابت يستخدم لتحديد عدد المواقع المطلوبة للعمل ويتم الوصول إليها من خلال الدليل أو الفهرس.
ب - التجانس: يستخدم لتمثيل نوع واحد من البيانات (نمط واحد).
ج- مستمرة: نطاق من الذاكرة مستمر (متماس).

من أهم ميزاتها التعريف بالجملة والاحتفاظ بالقيم للاستخدامات اللاحقة

4-1-المصفوفات وحيدة البعد

one-dimensional array

تسمى نسق أو صف وهو تتابع من الأهداف أو العناصر كلها من نفس النوع، ويتم ترقيمها بالتتابع من الصفر حتى n-1 لعدد n منها ويعرف بالفهرس index أو الدليل.
إذا كان اسم النسق a فإن a[0] هو اسم أوعنوان العنصر الموجود في المكان رقم صفر، في مثالنا (العنصر الأول 11.11)، و a[1] هو عنوان العنصر الموجود في المكان رقم 1 (العنصر الثاني 33.33) وهكذا، وبصورة عامة فإن العنصر a[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]

	a[0]	a[1]	a[2]	a[3]
a[ ]	11.11	33.33	55.55	77.77

4-1-1 الإعلان عن النسق.
Declaring of array

الإعلان عن النسق في لغة C++ يعني تحديد اسم النسق نوع العناصر وعددها. ليقوم المترجم بحجز المكان الكافي ضمن الذاكرة لهذا النسق.

الشكل العام:

elementType arrayName [array-size];

elementType: نوع العناصر التي سيخزنها النسق.

arrayName: اسم النسق.

array-size:قيمة ثابته تحدد حجم النسق أو عدد عناصره.

الإعلان ; double a[4] يعلن عن نسق اسمه a، عدد عناصره 4، نوعها حقيقي.

بل الاصح أن نكتب:

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

4-1-2- إدخال العناصر للمصفوفة توجد لها ثلاث طرق:

1- الإسناد أو التخصيص:تسند قيم ابتدائية لعناصر النسق.

2- الحلقات:تمر لكل موقع في المصفوفه وتضع قيمة فيه.

4-1-2- إدخال وإخراج عناصر النسق
Input, output the array elements

3- ادخال العناصر بذكر اسم كل منها، تقبلها لغة البرمجة ولاتستخدم لإنتهاك منطق المصفوفة.

الأولى:يمكن تخصيص قيم ابتدائية للنسق باستخدام قائمة تخصيص Customize list وتحصر عناصرها بقوسي مجموعة وتفصل عن بعضها بعضاً بفاصه عادية ;{ } ومثالاً.

```
float a[ 4 ] = { 22.2 , 44.4 , 66.6 , 88.8 } ;
```

نسف فيه 4عناصره محرفيه. ; char y[4]={'a','b','c','d'}

عند اسناد القيم يمكن حذف الحجم 4 وعندها يحدد بعدد القيم الموجودة في قائمة الاسناد.

ملاحظة 1: عند تحديد الحجم فإن عدد العناصر المسنده يمكن أن يكون أقل من 4 عندها يكمل الباقي أصفاراً.

2: عدد العناصر أكبر من الحجز 4 لاتقبل ويعطي خطأ في بناء الجملة syntax Error.

- طرق الإخراج هي نفس طرق الادخال (ما عدا الاسناد تستخدم للدخل فقط) مع استبدال تعليمة <<cin بـ <<cout.

4-1-2- إدخال وإخراج عناصر النسق

الطريقة الثانية: حلقة تمر على كل العناصر وفي كل مرور يتم إدخال قيمه للموقع الذي يشير له الدليل المعني مثالها:

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

أما الإخراج فيكون بالشكل :

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

ملاحظة 1: في الإخراج يجب اخراج فراغ بعد كل عنصر أو الانتقال لسطر جديد وإلا سيظهر الخرج وكأنه رقما واحداً.

ملاحظة 2 : يمكن استخدام أية حلقة
(for, while, do while)

~~الطريقة الثالثة:~~
~~ادخال العناصر بذكر اسم كل منها، تقبلها لغة البرمجة ولاتستخدم لإنتهاك منطق المصفوفة والدلالة على عدم خبرة وامكانية كاتب البرنامج.~~

```
{ cin>>a[0]>>a[1]>>a[2] >>a[3];}
 {cout<<a[0] << a[1]<< a[2]<< a[3];}
```

مثال 4- 1 :تطبيقات
الادخال والإخراج

ليكن المطلوب إدخل أربعة قيم وطباعتها بعكس ترتيب إدخالها.

```
void 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;
}
```

الخرج :

```
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
a[0]=4
```

مثال 4- 3 :تطبيقات
الادخال بالاسناد

ليكن المطلوب اسناد قيمتين لمصفوفة من 4 عناصر وملاحظة النتائج بعد طباعتها.

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

الخرج :

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

ملاحظة 1 : عند الحجز وعدم الادخال أو الاسناد وإخراج ما في المصفوفة ستكون النتيجة قيم عشوائية شأنها كحجز متغير وإخراج قيمته دون اسناد قيمه له مثلاً:

```
a[0]=2.97177e+175
a[1]=2.5855e-303
a[2]=3.17941e-303
a[3]=5.45306e-304
Press any key to continue . . .
```

4-1-4 تطبيقات على المصفوفات 1

حساب متوسط عناصر مصفوفة.

```
int main()
{   const int arraySize = 5;
   int a[arraySize];    int ave,s=0;
   for (int i = 0; i < arraySize ; i++)
        {cin>>a[i];              s += a[i];  }
    ave=s / arraySize;
   cout << "the average is " << ave << endl;
   return 0;
}
```

الخرج :

```
55
66
77
88
99
the average is 77
Press any key to continue . . .
```

4-1-4 تطبيقات على المصفوفات 2

البحث عن عنصر من عناصر مصفوفة ومعرفة موقعه إن وجد.

```
int main()
{   const int arraySize = 5;
   int a[arraySize];
  cout<<"input array elemant \n";

   for (int i = 0; i < arraySize ; i++)cin>>a[i];
   int pos=-1,item;
  cout<<"Enter the search value ";cin>>item;

   for (int j = 0; j < arraySize ; j++)
   if(a[j]==item){pos=j;}

   if(pos==-1)cout<<"\n item is not found\n";
   else cout<<"\n is found at "<<pos<<endl;
   system ("pause"); return 0;
}
```

البحث عن العنصر الذي يملك القيمة العظمى في النسق وموقعه.

```
int main()
{   const int arraySize = 5;
   int a[arraySize]; cout<<"input elemant\n";
   for (int i = 0; i < arraySize ; i++)cin>>a[i];
   int pos=0,max=a[0];
   for (int j = 1; j < arraySize ; j++)
   if(a[j]>max){max=a[j], pos=j;}
cout << "\nthe max values is " << max;
cout << "\nfound at pos " << pos<< endl;
   system ("pause"); return 0;
}
```

برنامج لإخراج قيم عناصر مصفوفة على شكل مخطط بياني histograms.

الخرج :

```
Element      Value      Histogram
      0         19       ********************
      1          3       ***
      2         15       ***************
      3          7       *******
      4         11       ***********
      5          9       *********
      6         13       *************
      7          5       *****
      8         17       *****************
      9          1       *
Press any key to continue . . .
```

```
void main()
{ const int arraySize = 10;
  int n[arraySize]={19,3,15,7,11,9,13,5,17,1};
  cout << "Element" << setw(13) << "Value"
       << setw(17) << "Histogram" << endl;
for (int i = 0; i < arraySize ; i++) {
    cout<<setw(7)<<i<<setw(13)<<n[i]<<"  ";
   for(int j = 1; j <= n[i]; j++)   // print one bar
        cout << '*';   cout << endl;
  }
}
```

حشر عنصر ضمن موقع محدد في مصفوفه

يتم إدخال موقع الحشر Position وقيمته Value

```
void main()
{
int a[ ] = {6,7,8,9,10};
int position,value,i;
for(i = 0; i < 5; i++) cout<<a[i]<<endl;
cout<<"Enter position [0 , 4] : "; cin>>position;
cout<<"Enter value to insert : ";cin>>value;
for(i=4; i>position; i--) a[i]=a[i-1];
 a[position]=value;cout<<"the new array is : "
<<endl;
for(i = 0; i < 5; i++) cout<<a[i]<<endl;
}
```

تم تصحيح الكود

الخرج :

```
6
7
8
9
10
Enter position [0 , 4] : 3
Enter value to insert : 333
the new array is :
6
7
8
333
9
Press any key to continue . . .
```

2-4- المصفوفات متعددة الأبعاد
Multiple subscripts arrays

4-2-1- الإعلان عن المصفوفة متعددة الأبعاد

يعلن عن المصفوفة متعددة الأبعاد بالشكل :

elementType arrayName [size1] [size2] .... [sizen]

مثال : double a [32] [10] [4] ; تم تعرف مصفوفة ثلاثية الأبعاد أبعادها: 4, 10, 32 .

إسناد قيم ابتدائية لعناصر المصفوفة yأثناء التصريح

$$y = \begin{bmatrix} 5 & 0 & -4 \\ -2 & 3 & 1 \\ 4 & 7 & 6 \\ 9 & 8 & -1 \end{bmatrix}$$

```
int y[4][3]={{5,0,-4},{-2,3,1},{4,7,6},{9,8,-1}};
```

حيث يتم تجميع عناصر كل صف ضمن قوسين.

والشكل التالي يوضح مصفوفة ذات بعدين:

	colum 0	colum 1	colum 2	colum 3
row 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
row 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
row 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

Colum subscript
Row subscript
Array name

فمثلاً التصريح:

```
int b[2][2]={{1},{3,4}}
```

يعطي العنصر b[0][0] القيمة 1 والعنصر b[1][0] القيمة 3 والعنصر b[1][1] القيمة 4 وباقي العناصر القيمة 0

4-2-2- إدخال المصفوفة وإخراجها متعددة الابعاد.

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

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

الفرق كما ذكرنا استبدال تغبسمة الدحل بتعليمة الخرج.

مثال 4-11 :

أكتب 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- طباعة مجموع عناصر المصفوفة.

```
void main()
{ int a[3][3]; int i,j,s=0;
for(i=0;i<3;i++) {cout<<i<<": ";
for(j=0;j<3;j++)cin>>a[i][j];}
cout<<"a array:"<<endl;
for(i=0;i<3;i++){
for(j=0;j<3;j++)cout<<a[i][j]<<" "; cout<<endl; }
cout<<endl;
```

```
cout<<"the elements of the main diameter
are:"<<endl;
for(i=0;i<3;i++)
for(j=0;j<3;j++)if(i==j)cout<<a[i][j]<<" ";
cout<<endl<<endl;
cout<<"the elements of the sub diameter
are:"<<endl;
for(i=0;i<3;i++)
for(j=0;j<3;j++)if(i+j==2)cout<<a[i][j]<<" ";
cout<<“\n\n”;
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;  }
```

بعض الأسئلة المطلوب الإجابة عنها -1-

1- المطلوب ذكر اهم مميزات المصفوفات وخدماتها.
2- أكتب برنامجاً يسمح بإدخال عناصر مصفوفة أحادية البعد مؤلفة من 50 عنصر، ومن ثم حشر عنصر في مكان ما. ملاحظة: يتم إدخال القيمة المراد حشرها وموقع الحشر من قبل المستخدم.
3- أعد كتابة البرنامج السابق بحيث يتم إدخال عناصر المصفوفة أحادية البعد المؤلفة من 50 عنصر ومرتبة من الصغير إلى الكبير، حشر العنصر في المكان الملائم له ضمن المصفوفة المرتبة.
ملاحظة: يتم تحديد موقع الحشر من قبل البرنامج.
4- أكتب برنامجاً يسمح بإدخال عناصر مصفوفة أحادية البعد مؤلفة من 50 عنصر، ومن ثم حذف عنصر من مكان ما وإزاحة باقي العناصر ومن ثم طباعة المصفوفة.
ملاحظة: يتم إدخال موقع الحذف من قبل المستخدم.
5- أكتب برنامجاً يستخدم ثلاث مصفوفات أحادية البعد CURRENT، RESISTANCE ، VLOT لتخزين قيم التيارات، المقاومات، الجهود على التوالي.
قم بإدخال قيم للمصفوفتين CURRENT ، RESISTANCE ومن ثم حساب قيم المصفوفة VOLT حيث أن:
VOLT[i] = CURRENT[i] * RESISTANCE[i]
قم بطباعة قيم المصفوفات الثلاث.
ملاحظة: يجب أن تكون المصفوفات الثلاث من نفس البعد

بعض الأسئلة المطلوب الإجابة عنها -2-

6- 1- قام مدرس مادة البرمجيات بإجراء امتحان لطلاب الصف البالغ عددهم 50 طالباً وبعد أن قام بتصحيح الأوراق أراد أن يجري العمليات التالية على علامات الطلاب ضمن الصف: 1- إدخال قيم علامات الطلاب.2- طباعة قيم علامات الطلاب في المادة. 3- إيجاد أعلى علامة وأدنى علامة وموقعهما. 4- حساب نسبة النجاح في المادة (علامة النجاح 60). 5-هناك معيار في تقييم نتائج مادة يعتبر أنه إذا كان أكثر من نصف الطلاب قد حصلوا على علامة أكبر من متوسط العلامات فإن النتائج منطقية. اختبر فيما إذا كانت علامات الطلاب في المادة منطقية.

6-2- هناك معيار أخر في تقييم نتائج مادة يعتبر أنه إذا كان مجموع انحرافات علامات الطلاب عن العلامة الوسطية موجباً فإن النتائج منطقية. اختبر فيما إذا كانت علامات الطلاب في المادة منطقية. 7- إنشاء نسخة من علامات الطلاب بحيث يتم استبدال كل علامة ناجحة بالقيمة 1 وكل علامة راسبة بالقيمة 0. -طور البرنامج السابق ليتعمل مع عدة مواد وعدة طلبة ويحسب متوسطات المواد ومتوسطات علامات الطلبة

8- يراد تخزين النتائج الامتحانية لثلاثين طالباً في 7 مواد مع معدلاتهم وأرقامهم الجامعية في مصفوفة ثنائية البعد بحيث أن العمود الأول يحوي قيمة الرقم الجامعي والأعمدة السبعة التالية تحوي درجات المواد والعمود الأخير يحوي المعدل والمطلوب:

إدخال القيم إلى المصفوفة الرقم الجامعي والعلامات.

حساب معدلات الطلاب وتخزينها في الأماكن الملائمة.

البحث عن رقم الطالب الذي حصل على أعلى معدل.

تحديد الطلاب غير الموفقين وهم الحاصلين في خمس مواد أو أكثر على علامة أقل من 60.

4-4-1-4- لعبة النرد

iomanip ملف رأسي يتضمن توابع منها:
setw(x) لترك x من الفراغات تكتب مرة في تعليمة الخرج.
setfill('0') لتعبئة الفراغات بالمحرف 0 تكنب لكل خرج.

-التابع ;( )rand : عند ندائه يعيد قيمة عشوائية ضمن المجال 0 to RAND_MAX والقيمة العظمى هي ثابت معرف في الملف الرأسي <cstdlib>.
-نأخذ باقي القسمة على 6 ونجمع عليه 1 فتصبح قيم عشوائية في المحال [0, ... , 6] وهي تعبر عن أوجه حجر النرد.
-حلقة تتكرر 20 مرة ينادى التابع ضمنها ويطبع مايعيده وننتقل لسطر جديد كل خمس طباعات (خمس قيم) if(i%5==0)cout<<"\n"; الأمر يكافئ رمي الحجر 20 مرة ونخرج كل 5 قيم على سطر.

```
int main()
{
  // loop 20 times
  for ( int coun= 1; coun<= 20; coun++ )
 {
  // pick rand number from 1 to 6 and print it
     cout << setw( 10 ) << ( 1 + rand() % 6 );
     // if coundivisible by 5, begin new line
     if ( coun% 5 == 0 )       cout << endl;
  } // end for structure
  system ("pause");
 return 0; // indicates successful termination
} // end main
```

// Roll a six-sided die 6000 times.

```
void main() {
   int fre1=fre2 = fre3 =fre4 =fre5=fre6 = 0;
   int face;  // represents one roll of the die
    // loop 6000 times and summarize results
   for ( int roll = 1; roll <= 6000; roll++ ) {
     face = 1 + rand() % 6;
      switch ( face ) {
        case 1:    ++fre1;      break;
        case 2:    ++fre2;      break;
        case 3:    ++fre3;      break;
        case 4:    ++fre4;      break;
        case 5:    ++fre5;      break;
        case 6:    ++fre6;      break;
        default: cout << "Program should never get here!";
     } // end switch
   } // end for
```

```
cout << "Face" << setw( 13 ) << "fre"
 << "\n   1" << setw( 13 ) << fre1
 << "\n   2" << setw( 13 ) << fre2
 << "\n   3" << setw( 13 ) << fre3
  << "\n   4" << setw( 13 ) << fre4
   << "\n   5" << setw( 13 ) << fre5
    << "\n   6" << setw( 13 ) << fre6 << endl;
   system ("pause");
} // end main
```

الخرج مثلاً

```
 Face        fre
   1        1003
   2        1017
   3         983
   4         994
   5        1004
   6         999
Press any key to continue . . .
```

srand (time (0));

```
cout << "Face" << setw( 13 ) << "fre"
 << "\n   1" << setw( 13 ) << fre1
 << "\n   2" << setw( 13 ) << fre2
 << "\n   3" << setw( 13 ) << fre3
  << "\n   4" << setw( 13 ) << fre4
   << "\n   5" << setw( 13 ) << fre5
    << "\n   6" << setw( 13 ) << fre6 << endl;
   system ("pause");
} // end main
```

الخرج مثلاً

```
 Face        fre
   1        1003
   2        1017
   3         983
   4         994
   5        1004
   6         999
Press any key to continue . . .
```