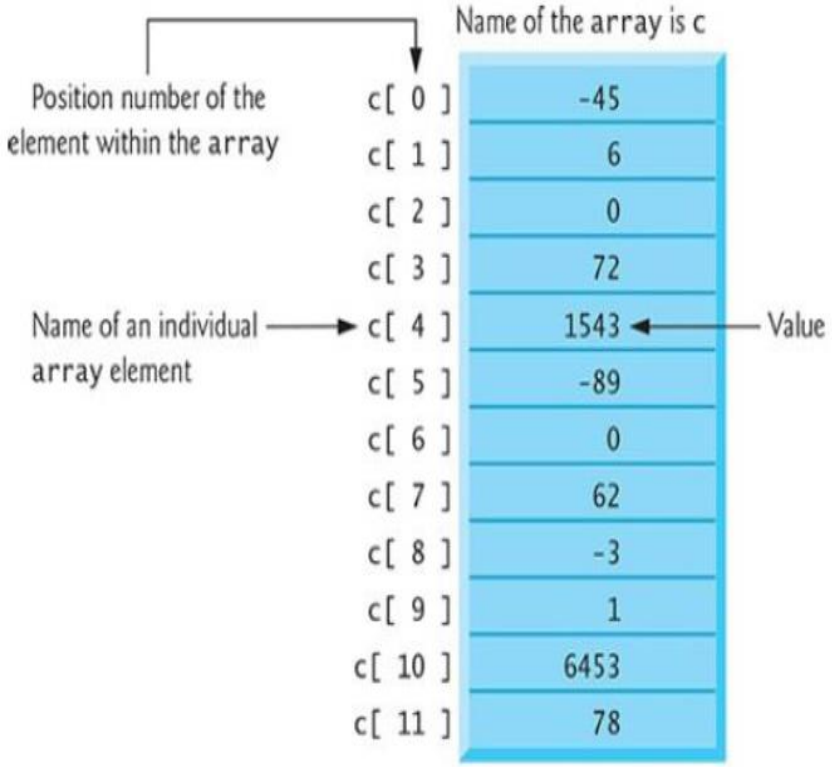
برمجة 1 introduction to programming

1. المصفوفات Arrays
2. المصفوفات والسلاسل
3. التوابع function
4. التوابع العودية
5. المؤشرات pointers
6. المؤشرات والمصفوفات
7. السجلات Struct
8. الصفوف Classes
9. التابع الباني constructed function
10. التابع الهادم destroyed function
11. البرمجة غرضية التوجه Object oriented programming

المحاضرة الأولى د. كندة سليمان أبوقاسم

- Example:

```
double C[12];
```



Array.. المصفوفات

المصفوفات هي نوع معطيات جديد يستخدم لتخزين قيم متعددة ضمن اسم واحد, وهي بنية معطيات مركبة ولكن جميعها تملك نفس نوع البيانات

. المصفوفات أحادية البعد One dimensionsl array
تسمى المصفوفة وحيدة البعد أو الأنساق لها نفس نوع البيانات , تسمى عناصر النسق, ويتم ترقيمها بالتتابع من الصفر
indexوهذه الأرقام تسمى بالفهرس من العناصر أي ...3,2,1,0 حتى n-1
تتطلب لغة C++ أن يكون حجم النسق عدد صحيح موجبsize-array
فالاعلان :

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

يمكن تعريف المصفوفة بالشكل التالي `int b [] = {11, 45, 62, 70, 88};`
حيث تتوسع المصفوفة بعدد العناصر المدخلة

الإعلان عن المصفوفة أحادية البعد .

Declaring array

يتم بتحديد نوع العناصر وعددها لكي يقوم المترجم بحجز المكان الكافي ضمن الذاكرة لهذا النسق ويكون بالشكل التالي :

Type array-name [array-size] ;

حيث:

نـوع عناصر النسق : type

عدد عناصر النسق : size-array

name-array :

data type arrayName[Size];

The Size must be an integer constant greater than zero.

For example:

```
int a[10];
char name[20];
float temperature[6];
```

Accessing array elements:

arrayName[index]

- All arrays have 0 as the index of their first element and Size-1 as the index of their last element.
- The arrayName represents the address of the first element in the array.

For example:

int a[10];

The first element is a[0]

The last element is a[9]

The array is a[0] , a[1] , a[2] , . . . , a[9]

هذا ويمكن أن يعلن عن النسق السابق بالشكل :

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

sizeحيث أعلن بداية عن على أنه ثابت صحيح و قيمته 4

إدخال و إخراج عناصر النسق
Input and output the array elements

يتم إدخال عناصر النسق الذي أعلن عنه سابقاً بالشكل باستخدام الحلقات التكرارية

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

أما الإخراج فيكون بالشكل

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

قراءة مصفوفة من لوحة المفاتيح

```
Example:
int marks [10];
    for(int i=0;i<10;i++)
     {
      cin>>marks[i];
     }
```

مثال حساب مجموع عناصر مصفوفة

```
int arr [5] = {11, 35, 62, 476, 989};
int sum = 0;

for (int x = 0; x < 5; x++) {
sum += arr[x];
}
cout << sum << endl;
//Outputs 1573
```

```
char name[6]={'G','A','U','R','A','V','\0'};
```

مثال اكتب برنامج لإدخال 5 ارقام وتخزينها في مصفوفة ثم يطبع عناصر المصفوفة

```
#include <iostream>
using namespace std;

int main()
{
    int numbers[5];
    cout << "Enter 5 numbers: " << endl;
      //  store input from user to array
     for (int i = 0; i < 5; ++i) {
            cin >> numbers[i];
       }
      cout << "The numbers are: ";
       //  print array elements
      for (int n = 0; n < 5; ++n) {
       cout << numbers[n] << "  ";
      }
  return 0;
}
```

Output

```
Enter 5 numbers:
11
12
13
14
15
The numbers are: 11  12  13  14  15
```

مثال

اكتب برنامج حساب متوسط عناصر مصفوفة.

Calculate Average Array element

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

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

البحث الخطي

مثال اكتب برنامج للبحث عن قيمة مدخلة من لوحة المفاتيح في مصفوفة أعداد صحيحة مؤلفة من خمس عناصر

```
/ / Linear search
# include < iostream.h >
void main ( )
{
        int A[5], i, data, flag = 0;
        cout < < "Enter five values";
        for (i = 0; i < 5; i++)
        cin > > A [i];
        cout << 'Enter data to be searched";

        cin >> data;
        for (i=0; i < 5; i ++)
                {
                if (A[i] = = data)
                flag = 1;
                }
        if (flag = = 1)
        cout < < "Data present";
        else cout << "Data not present";
}
```

Output of the program

```
Enter five values
34
87
67
56
12
Enter data to be searched 67
Data present
```

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
 {const int size=10;
 int a[size]={19,3,4,8,10,12,20,5,33,12} ;

 // loop 10 times and calculate and output square of x
each time
 cout <<"element "<<setw(13)<<"value"
<<setw(17)<<"histogram" <<endl;
 for (int i = 0;i < size;i++){

 cout <<setw(7)<<i<< setw(13)<<a[ i]<<"   "  ;
     for (int j=0;j<a[i];j++)
                  cout<<"*";
              cout << endl;
 }
}
```

مثال

اكتب برنامج لطباعة histogram يقوم بطباعة عناصر مصفوفة من 10 عناصر على شكل بياني حيث تمثل قيمة العنصر عدد الرموز النجمية في السطر

```
element       value      histogram
      0          19  ********************
      1           3  ***
      2           4  ****
      3           8  ********
      4          10  **********
      5          12  ************
      6          20  ********************
      7           5  *****
      8          33  *********************************
      9          12  ************
```

الفرز الفقاعي Bubble sort

تعتمد طريقة الفرز الفقاعي على القيام بأكثر من مرور على عناصر النسق .

في كل مرور يتم مقارنة زوجين متتالين من عناصر النسق, إذا كان هذان الزوجان مرتبين تصاعدياً (أو متساويين (فإننا نبقيهما على حالهما,

أما إذا كانا مرتبين تنازلياً فإننا نقوم بالمبادلة بينهما ضمن النسق (فرز فقاعي تصاعدي)

```
#include <iostream>
using namespace std;
int main()
{
   const arraysize=5;
    int a[arraysize]={10,8,6,4,2};
    int hold;
    cout<<"data items in original order:\n";
     for(int i=0;i<arraysize;i++)
           cout<<a[i]<<" ";
        for(int pass=1 ;pass<arraysize;pass++)
        for(i=0;i<arraysize-1;i++)
            if(a[i]>a[i+1])
         { hold=a[i]; a[i]=a[i+1]; a[i+1]=hold; }
        cout<<endl<<"data items in ascending order:\n";
        for(i=0;i<arraysize;i++)
        cout<<a[i]<<" ";
   return(0);
}
```

Output

data items in original order:

10 8 6 4 2

data items in ascending order:

2 4 6 8 10

pass	i	a[0]	a[1]	a[2]	a[3]	a[4]
1	0	8	10	6	4	2
	1	8	6	10	4	2
	2	8	6	4	10	2
	3	8	6	4	2	10
2	0	6	8	4	2	10
	1	6	4	8	2	10
	2	6	4	2	8	10
	3	no change				
3	0	4	6	2	8	10
	1	4	2	6	8	10
	2	no change				
	3	no change				
4	0	2	4	6	8	10
	1	no change				
	2	no change				
	3	no change				

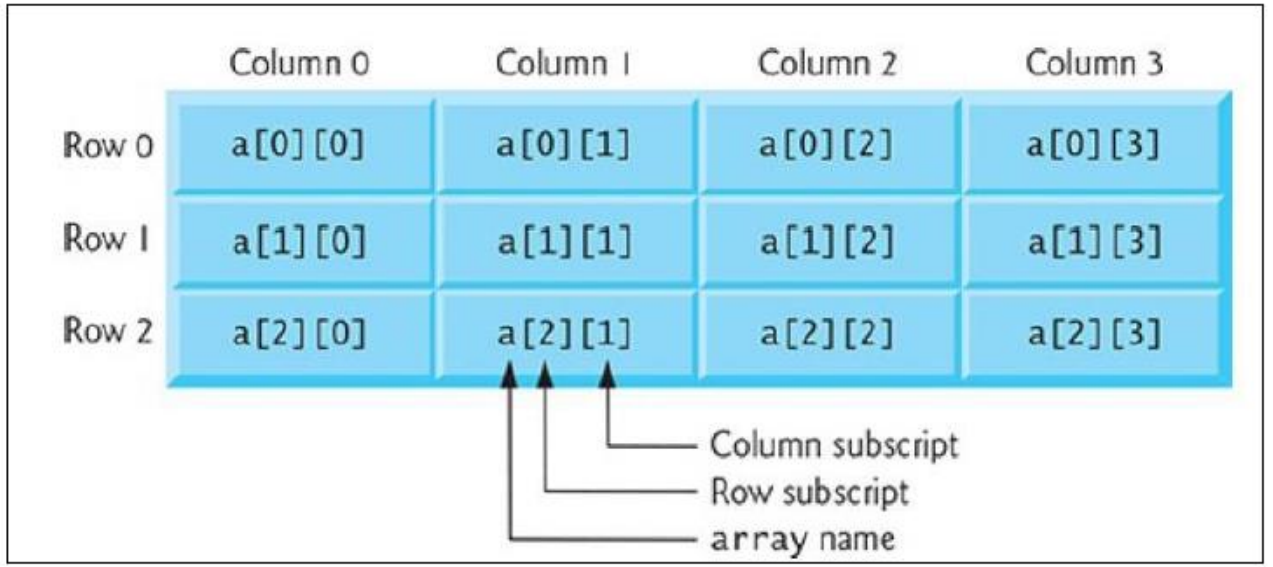
المصفوفة ثنائية البعد Multidimensional Arrays

الشكل العام لتعريف المصفوفة ثنائية البعد

Declaring Two Dimensional Arrays

- Type array_name[rows][columns];

- Example:
- int marks[25][7] //marks is a 2D array of 25 rows and 7 columns of int values



طباعة عناصر مصفوفة ثنائية البعد

يتم طباعة عناصرمصفوفة ثنائية البعد a[4][3]باستخدام الحلقات التكرارية

```
for ( int i = 0 ; i < 4 ; i++ ){
      for(int j=0; j< 3 ; j++){
      cout >> a[i] [j] ;}
}
```

يتم إدخال عناصرمصفوفة ثنائية البعد a[4][3] الذي بالشكل باستخدام الحلقات التكرارية

```
for ( int i = 0 ; i < 4 ; i++ ){
      for(int j=0; j< 3 ; j++){
      cin >> a[i] [j] ;}
}
```

مثال

اكتب برنامج لاملاء مصفوفة اعداد صحيحة ب أصفار ثنائية البعد مؤلفة من 25 سطر وسبع اعمدة وطباعة عناصر

```
#include<iostream>
using namespace std;
int main()
 {
 int marks[25][7]; //marks is a 2D array of 25 rows and 7 columns of int values
      for(int i = 0; i<25; i++)
            {
                  for(int j = 0; j<7; j++)
                   {
                           marks[i][j] = 0;
                   }//for j
            }//for i
     cout<<”Array contents\n”;
            for(int i = 0; i<25; i++)
            {
                  for(int j = 0; j<7; j++)
                   {
                           cout<<marks[i][j]<<”\t”;
                   }
                  cout<<”\n”;
            }
  }//main()
```

```
Array contents
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0
0       0       0       0       0       0       0

--------------------------------
```

مثال اكتب برنامج لقراءة وطباعة عناصر مصفوفة اعداد صحيحة ثنائية البعد مؤلفة من 25 سطر وسبع اعمدة

```
#include<iostream>
using namespace std;
int main()
 {
 int marks[25][7]; //marks is a 2D array of 25 rows and 7 columns of int values

        for(int i = 0; i<25; i++)
        {
                for(int j = 0; j<7; j++)
                 {
                        cin>>[i][j];
                 }//for j
        }//for i
  cout<<"Array contents\n";
        for(int i = 0; i<25; i++)
        {
                for(int j = 0; j<7; j++)
                 {
                        cout<<marks[i][j]<<"\t";
                 }
                cout<<"\n";
        }
}//main()
```

مثال اكتب برنامج لطباعة عناصر مصفوفة اعداد صحيحة ثنائية البعد مؤلفة من سطرين وثلاث اعمدة

```
#include<iostream>
 using namespace std;
 int main()
 {
 int x[2][3] = {{2, 3, 4}, {8, 9, 10}};

for(int i = 0; i<2; i++)
        {
                for(int j = 0; j<3; j++)
                 {
                        cout<<x[i][j]<<"\t";
 }
cout<<"\n";
}
}
```

مثال

اكتب برنامج يقوم بإدخال مصفوفة ثنائية من الاعداد الصحيحة حجمها 3 x5و المطلوب:

1 - طباعة المصفوفة .

2 - طباعة مجموع عناصر المصفوفة .

3 - طباعة العناصر التي ًكون دليل العمود فيها أكبرمن دليل السطر

```
#include <iostream>
using namespace std;
int main()
{
int x[3][5];
 for(int i=0;i<3;i++){
      for(int j=0;j<5;j++)
      cin>>x[i][j];}
for(int i=0;i<3;i++) {
    for(int j=0;j<5;j++){
   cout<<x[i][j]<<" "; }
cout<<endl; }
int sum=0;
 for(int i=0;i<3;i++) {
       for(int j=0;j<5;j++) {
        sum=sum+x[i][j]; } }
  cout<<"sum="<<sum<<endl;
 cout<<"the elements which column index is greater than row index
:"<<endl;
 for(int i=0;i<3;i++) {
  for(int j=0;j<5;j++) {
      if(j>i)
      cout<<x[i][j]<<" " ;
      else
      cout<<'0'<<"  "; }
  cout<<endl; }
  return 0; }
```

طريقة البحث الثنائي Binary search

تعتمد طريقة البحث الثنائي على

1- يجب أن تكون المصفوفة مرتبة تصاعديا

2- تقسيم المصفوفة على 2

3- حذف نصف عدد عناصر النسق المرتب الذي نبحث ضمنه بعد كل عملية مقارنة,

4- حيث تقوم خوارزمية هذه الطريقة بمقارنة قيمة العنصر الواقع في منتصف النسق مع القيمة المفتاح التي نبحث عنها, فإذا تساوت القيمتان فهذا يعني أنه تم إيجاد القيمة التي نبحث عنها واذا لم تحدث المساواة فيتم إعادة صياغة المسألة لتصبح مكافئة لمسألة إيجاد عنصر ضمن أحد نصفي النسق السابق

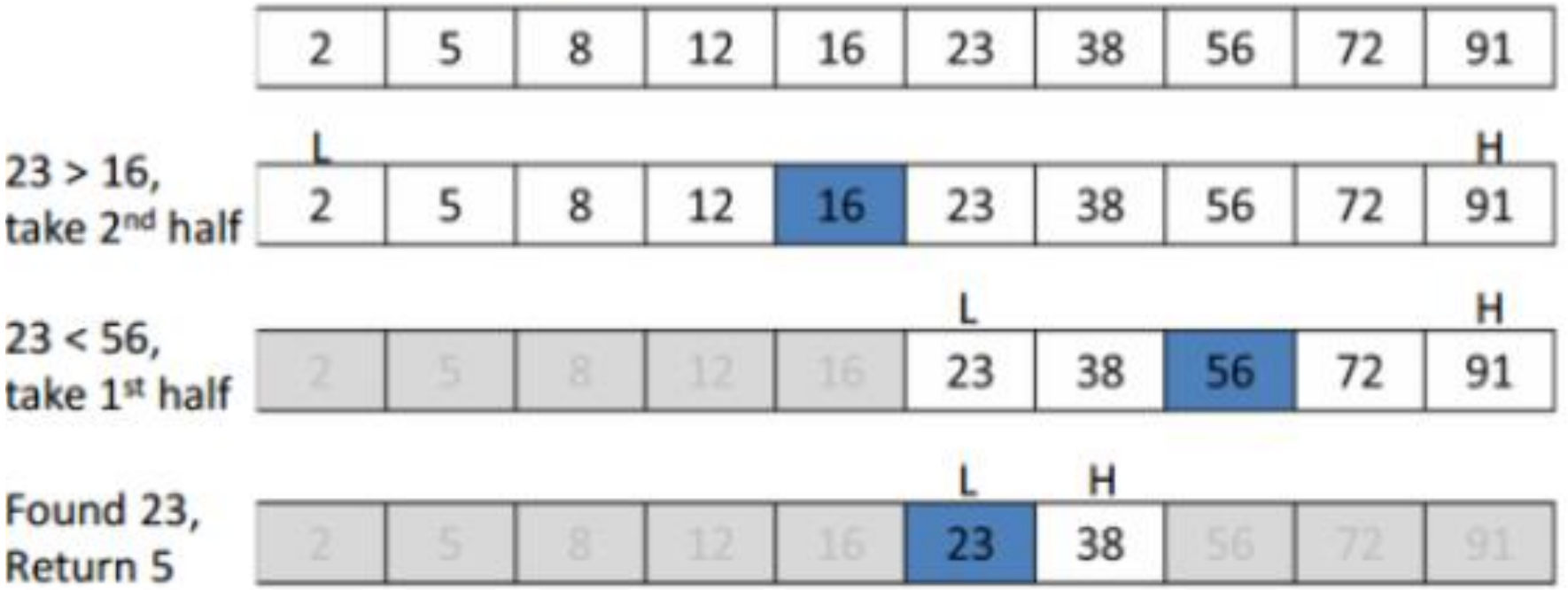
5- إذا كانت القيمة المفتاح تقل عن قيمة العنصر الواقع في منتصف النسق فإنها تقع في النصف القيمة التي نبحث عنها تقع في النصف الأول من النسق وذلك على افتراض أن عناصر النسق مرتبة تصاعدياً.

إذا لم تكن القيمة المفتاح مساوية لقيمة العنصر الوسط في النسق الجزئي المختار نتيجة المرحلة الأولى فإن الخوارزمية تتكرر على أحد أرباع النسق الأصلي وهكذا.

في أسوأ الحالات تقوم خوارزمية البحث الثنائي بعشر مقارنات للبحث عن عنصر ضمن مصفوفة مؤلفة من 1024 عنصرًا, إذ أن التقسيم المتكرر على 2 يحذف نصف عدد العناصر بعد كل عملية مقارنة .

(1 , 2 , 4 , 8 , 16 , 32 , 64 , 128 ,256 , 512)

If searching for 23 in the 10-element array



Left=0 t=23
Right=n-1

Location=(left + right)/2
Loc=4

Left=location +1
Right=n-1
Location=(left + right)/2
Loc=(5+9)/2=7

يوضح هذا البرنامج طريقة البحث الثنائي عن عنصر في صف

هذا البرنامج يسمح بالبحث عن القيمه الهدف target حتى يتم إدخال الرقم 0 في حال وجدت القيمة الهدف يطبع فهرسها والا تطبع رسالة تعلن عن عدم إيجادها.

```
#include<iostream.h>
void main() { const int n=8; int target;
int a[n]={22,33,44,55,66,77,88,99}; cout<<"a array :";
for(int i=0;i<8;i++)
cout<<a[i]<<" "; cout<<endl;
do{ cout<<"enter target,0 to end:";
cin>>target;
 int location,left=0, right=n-1; int found=0;
while(!found && left<=right){
location=(left+right)/2;
found=(a[location]==target);
if(a[location]<target)
eft=location+1;
else right=location-1; }
if(found)cout<<target<<"is at:"<<location<<"\n";
if(!found)cout<<target<<"is not found.\n";
}
while(target!=0); }
```

```
a array : 22 33 44 55 66 77 88 99
enter target , 0 to end:55
55 is at : 3
enter target , 0 to end : 6
6 is not found.
enter target , 0 to end : 2
2 is not found.
enter target , 0 to end : 99
99 is at : 7
enter target , 0 to end :0
0 is not found.
Press any key to continu
```

المصفوفات والسلاسل Arrays and Strings

تمثيل المصفوفات في الذاكرة :
في C ++، يتم تعيين أي مصفوف إلى موقع ذاكرة قريب .
جميع عناصر الذاكرة تتواجد بجانب بعضها البعض.
يتوافق العنوان الأدنى مع العنصر الأول والأعلى عنوان العنصر الأخير

Note: In C++ the first element has the index **zero**!

```
int part_numbers[] = {123, 326, 178, 1209};

int scores[10] = {1, 3, 4, 5, 1, 3, 2, 3, 4, 4};

 char alphabet[5] = {'A', 'B', 'C', 'D', 'E'};

int sample[10];

float float_numbers[100];

char last_name[40];
```

A static local variable التخزين الساكن للمصفوفات

وجد متغير محلي ثابت طوال مدة البرنامج ولكنه مرئي فقط في جسم التابع .
يمكننا تطبيق تعريف مصفوفة محلي staticإنشاء المصفوفة وتهيئتها في كل مرة تقوم فيها الدالة
يتم استدعاء المصفوفة ولا يتم إتلافها في كل مرة يتم فيها إنهاء الدالة في البرنامج.
يؤدي ذلك إلى تقليل زمن تنفيذ البرنامج ، خاصة بالنسبة للبرامج ذات الدوال التي يطلق عليها بشكل متكرر والتي تحتوي على مصفوفات كبيرة

automatic arrays

في الدوال التي تحتوي على مصفوفات تلقائية حيث تكون الوظيفة داخل النطاق وخارجه بشكل متكرر ، اجعل المصفوفة ثابتة بحيث لا يتم إنشاؤها في كل مرة يتم استدعاء الدالة
يتم تهيئة المصفوفات الثابتة بمجرد بدء تشغيل البرنامج.
إذا لم تقم بذلك صراحة تهيئة مصفوفة ثابتة ، يتم تهيئة عناصر هذه المصفوفة إلى الصفر افتراضيًا.

You cannot assign one array to another in C++.

The following is illegal:

```
int a[10], b[10];

// do something
// assign all elements of array b to array a
a = b; // error -- illegal
```

Instead, you have to do the assignments for each element:

```
int i;

// assign all elements of array b to array a
for(i=0; i<10; i++) a[i] = b[i];
```

Example:

```
int a[8];
int j;

for(j=0; j<8; j++) a[j] = 7-j;
```

Then the memory representation of array **a** looks like this:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
7	6	5	4	3	2	1	0

فحص الحدود Bounds Checking

لا يقوم C ++ بفحص أي حدود على المصفوفات.

لا شيء يمنعك من تجاوز نهاية المصفوفة:

سوف تقوم بتعيين قيم لبعض المتغيرات الأخرى البيانات

يمكنك حتى الكتابة في جزء من برنامج الشفرة

على سبيل المثال ، يمكنك ترجمة وتشغيل البرنامج التالي ، حتى على الرغم من تجاوز حدود المصفوفة:

```
// An incorrect program. Do not execute!
int main()
{
int crash[10], i;
for(i=0; i<100; i++)
  crash[i] = i;
return(1);
}
```

تسمح مصفوفات الأحرف التي تحتوي على سلاسل بتهيئة الاختزال يأخذ هذا الشكل:

```
char array-name[size] = “string”;
```

على سبيل المثال ، يقوم جزء التعليمات البرمجية التالي بتهيئة str إلى الامتداد عبارة “Hello”

```
char str[6] = “hello”;
```

هذا هو نفس الكتابة

```
char str[6] = {‘h’, ‘e’, ‘l’, ‘l’, ‘o’, ‘\0’};
```

تذكر أنه يجب التأكد من جعل المصفوفة طويلة بما يكفيل تضمين فاصل الصفر

للإعلان عن مصفوفة يمكن أن تحتوي على سلسلة مكونة من 10 أحرف ، واحد سيكتب

```
char str[11];
```

تحديد الحجم كـ 11 يفسح المجال للصفر في نهاية السلسلة

Example 1

Write a C++ program that finds the average of each row of a 3× 4 matrix input by the user.

```
#include <iostream>
using namespace std;
int main()
{
 int a[3][4];
 int sum;
 cout<<"Enter 3x4 integer matrix: ";
 for (int i=0; i<3; i++)
 for (int j=0; j<4; j++)
 cin>>a[i][j];
  for (int i=0; i<3; i++){
 for (int j=0; j<4; j++){
  cout<<a[i][j]<<"   ";}
 cout<<endl;}
  cout<<"Average of each row: "<<endl;
 for (int i=0; i<3; i++)
 {
sum = 0;
 for (int j=0; j<4; j++)
 sum += a[i][j];
 cout<<sum/4.0<<endl;
 }
}
```

Example 1

Write a C++ program that computes the number of even integer numbers in an array entered by the user.

```
#include <iostream>
using namespace std;
int main()
{
 const int size = 10;
 int a[size] , count = 0;
 cout<<"Enter ten integer numbers: ";
 for(int i=0 ; i<10 ; i++)
 {
 cin >> a[i];
 if(a[i] % 2 == 0)
 count++;
 }
 cout<<"The number of even numbers is " << count;
}
```

Write a C++ program that reads a string and then computes the number of capital letters in the string.

```
 main()
{
 char str[30];
 int count = 0;
cout<<"Enter your string: ";
 cin >> str;
 for(int i=0 ; str[i] ; i++)
 if(str[i] >= 'A' && str[i] <= 'Z')
 count++;
 cout<<"No. of capital letters is " << count;
}
```

1- Find and print the summation of the main diagonal of a matrix of size 10x10 of type integer

2- Find and print the summation of each column of 7x5 array of type integer.

3-Find and print the summation of each odd column in a 10x10 array of type integer.

4- Find and print the summation of each row of 7x5 array of type integer.

5- Find and print the summation of the elements that are above the main diagonal in a matrix of size 10x10 of type integer.

6- Find and print the summation of each odd column in a 10x10 array of type integer.

7- Find and print the summation of each even row in a 10x10 array of type float