## الفصل الخامس (Section 5)
## المصفوفات (Arrays)

### 1.5- مقدمة :-

أن طرق التعامل مع أسماء المتغيرات والثوابت العددية والرمزية ، التي وردت في الفصول السابقة ، تعد صالحة للتعامل مع عدد محدود من هذه الثوابت والمتغيرات ،سواء في عمليات الإدخال والإخراج أو في العمليات الحسابية والمنطقية ، وعندما يصبح عدد المتغيرات كبيرا جدا ، تصبح تلك الطرق غير عملية ، فمثلا لو أردنا إدخال مائة قيمة للمتغيرات x1, x2,....,x100، فكم الحيز المطلوب من البرنامج لعمليات الإدخال والإخراج والعمليات الحسابية والمنطقية لهذه المتغيرات ؟

هذا من جهة ، ومن جهة أخرى : فأننا نوفر مخزنا خاصا لك لمتغير نتعامل معه ،أثناء تنفيذ البرنامج ، ولذلك لحفظ قيمته في مخزن ، ومن ثم لاستعمال قيمته في عمليات أخرى تالية ، ومن ناحية ثالثة ، فان من الصعوبة بمكان ، بل من المستحيل استعمال اسم المتغير العددي أو الرمزي كمصفوفة ذات بعدين ، وثلاثة أبعاد... الخ.

للأسباب الواردة أعلاه ، جاءت فكرة استعمال متغير جماعي يضم تحت اسمه عددا من العناصر يسمى المصفوفة (Array)، المصفوفة هي نوع من أنواع بنية البيانات وهي مجموعة من المواقع المتجاورة في الذاكرة، لها عدد محدود ومرتب من العناصر التي تكون جميعها من نفس النوع type، فمثلاً يمكن أن تكون جميعها صحيحة int أو غير صحيحة float ولكن لا يمكن الجمع بين نوعين مختلفين في نفس المصفوفة .

الشكل التالي يبين مصفوفة C تحتوى على 13 عنصر من النوع int، ويمكن الوصول إلي أي من هذه العناصر بذكر اسم المصفوفة متبوعاً برقم موقع العنصر في المصفوفة محاطاً بالأقواس [ ] .

يرمز لرقم العنصر في المصفوفة بفهرس العنصر (index). فهرس العنصر الأول في المصفوفة هو 0 ولهذا يشار إلى العنصر الأول في المصفوفة C بـ C[0] والثاني C[1] والسابع C[6] وعموماً يحمل العنصر i في المصفوفة C الفهرس C[i-1]. تتبع تسمية المصفوفات نفس قواعد تسمية المتغيرات.

| | |
|---|---|
| C[0] | -45 |
| C[1] | 6 |
| C[2] | 0 |
| C[3] | 72 |
| C[4] | 1543 |
| C[5] | -89 |
| C[6] | 0 |
| C[7] | 62 |
| C[8] | -3 |
| C[9] | 1 |
| C[10] | 6453 |
| C[11] | 78 |
| C[12] | 15 |

أحياناً يسمى فهرس العنصر برمز منخفض subcript ويجب أن يكون الفهرس integer أو تعبير جبري تكون نتيجته integer . فمثلاً إذا كانت a=5 وb=6 فالعبارة: **C[a+b]+=2**
معناها نقوم بإضافة 2 إلي العنصر الثاني عشر C[11] في المصفوفة C.
يحمل العنصر 0 في المصفوفة C القيمة 45- والعنصر 1 القيمة 6. لطباعة مجموع الثلاثة عناصر الأولى في المصفوفة C يمكن كتابة:

```
cout<<C[0]+C[1]+C[2]<<endl;
```

**من الفوائد المهمة للمصفوفات** : هو استعمالها في الترتيب التصاعدي والتنازلي للعناصر والقيم المختلفة ، وعمليات ترتيب الأسماء الأبجدية او النصوص الرمزية ، وفي عمليات ضرب المصفوفات ، وإيجاد معكوس المصفوفة وعملياتها الأخرى ، وفي التحليل العددي ... الخ.

### 2.5- كيفية الإعلان عن المصفوفة(How to declare the array):-

تحتل المصفوفات حيزاً في الذاكرة لذا يجب على المبرمج تحديد نوع عناصر المصفوفة وعددها حتى يتسنى للمعرف تخصيص الحيز اللازم من الذاكرة لحفظ المصفوفة و الصيغة العامة لها هي

*data-type* array name of array [ *size* ];

نوع بيانات المصفوفة — اسم المصفوفة — حجم المصفوفة

### 3.5- تهيئة عناصر المصفوفة (Initializing Array Elements):-

عند اعلان مصفوفة فأنه ممكن أن نذكر نوع المصفوفة و اسمها و ابعاد المصفوفة دون تعيين قيم او محتويات للمصفوفة و في مثل هذه الحالة تكون قيمتها 0 كما في المثال الاتي:-

```
Int
a[20];
```

كذلك ممكن ان نعلن عن محتويات المصفوفة عند تعريفها و المثال التالي يوضح ذلك

```
int a [10] = { 8, 10, 13, 15, 0, 1, 17, 22};

int x [ ] = { 12, 3, 5, 0, 11, 7, 30, 100, 22 };

age [9] = { 18, 17, 18 ,18 ,19, 20 ,17, 18 ,19 };
```

### 4.5- كيفية الوصول إلى عناصر المصفوفة( How to Accessing Array Elements):-

نحن نصل إلى كل عنصر من عناصر المصفوفة بالاسم المكتوب للمصفوفة ، يليهاالأقواس التي تحدد المتغير (أو الثابت) في الأقواس التي تسمى فهرس المصفوفة وللوصول الى عناصر المصفوفة نقوم بتحديد موقع العنصرداخل المصفوفة والمطلوب الوصول إليه كما في الامثلة التالية التابع للمثال اعلاه:-

- The first element of array age:

```
age [0] = 18;
```

- The last element of array age:

```
age [9] = 19;
```

مثال اخر للمصفوفة اسمها num[10]

- Accessing the first element of array num to variable x:

x = num [0]; تعيين العنصر الاول من المصفوفة للمتغير x

- Accessing the last element of array num to variable y:

y = num [9]; تعيين العنصر الاول من المصفوفة للمتغير y

- cout << num [0] + num [9];

- num [0] = num [1] + num[2];

-num [7] = num [7] + 3; ⟷ num [7] += 3;

**5.5 - انواع المصفوفات (types of array):-**

و تقسم المصفوفات الى عدة انواع بالاعتماد على ابعاد المصفوفة و هي :-

1. **المصفوفة ذو البعد الواحد (One-dimensional Array):-**

هو مصفوفة ذات بعد واحد أو متجه (vector) ويمثل في الجبر على النحو الأفقي [ $a_1$ $a_2$ ...$a_n$ ] أو العمودي .

**الصيغة العامة لمصفوفة البعد الواحد هي :-**

*data-type* Array-name of array [ *size* ];

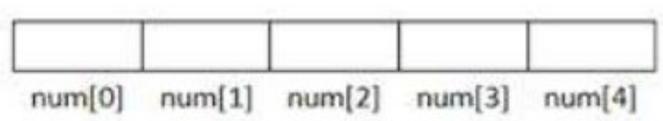
[ نوع بيانات المصفوفة] [ اسم المصفوفة][ حجم المصفوفة ]

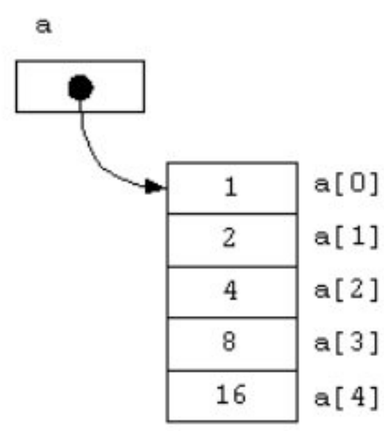
**Examples:**

int age [10];
int num [30];
float degree[5];
char a [15];

Example:

int num[5];




وحتى تخبر الذاكرة بأن يخصص حيزاً لـ 12 عنصر من النوع int في مصفوفة C ، استخدم الإعلان:

int C[12];

يمكن تخصيص الذاكرة لعدة مصفوفات باستخدام نفس الإعلان وذلك كالأتي:

```
int b[100], x[20];
```

أيضاً يمكن الإعلان عن مصفوفات من أي نوع بيانات آخر ، فمثلاً للإعلان عن مصفوفة عناصرها من النوع char نكتب:

```
char ch[20];
```

**إعطاء قيمة أولية للمصفوفة ذات البعد الواحد(Array Initialization):-**

مثال على إدخال عدة عناصر من مصفوفة الدرجات grade[ ]

```
int grade[5]={80,90,54,50,95};
```

ومثال أخر على إدخال قيم عناصر المصفوفة الرمزية name[ ]

```
Char name[4]="nor";
```

لاحظ أن المتغير المرقم name[ ] مكون من أربعة عناصر بينما تم إعطاؤه ثلاثة عناصر فقط والسبب أن العنصر الرابع بالنسبة إلى المعطيات الرمزية يكون خاليا.

```
#include < iostream.h>
main ()
{
int a[6]={40,60,50,70,80,90}
int I;
for(I=0;I<6;I++)
{
cout<<a[I]<<endl;
}
}
```

**Example 1:-** Write a program that finds a total, average student score in 5 subjects, and theseThe grades are as follows: 87,67,81,90,55?

```
#include < iostream.h>
main ()
{
int a[5]={87,67,81,90,55}
int s=0;
for(int i=0;i<5;i++)
s=s+a[i];
float avg=s/5;
cout<<avg<<endl;<<s<<endl;
}
```

**Example 2-:**Write C++ program to display 2nd and 5th elements of array distance?

```
#include<iostream.h>
void main( )
{
double distance[ ] = { 23.14, 70.52, 104.08, 468.78, 6.28};
cout << "2nd element is: " << distance[1] << endl;
cout << "5th element is: " << distance[4];
}
```

**Example 3:-**Write C++ program to read 5 numbers and print it in reverse order?

```
#include<iostream.h>
void main( )
{
int a [5];
cout << "Enter 5 numbers \n";
for ( int i =0; i <5; i++ )
{
cout << i << ": ";
cin >> a [ i ];
cout << "\n";
}
cout << "The reverse order is: \n";
for ( i =4; i >=0; i-- )
cout << i << ": " << a [ i ] << endl;
}
```

**Example 4:-**Write C++ program, to find the summation of array elements?

```
#include<iostream.h>
void main ( )
{
int const L = 10;
int a [L];
int sum = 0;
cout << "enter 10 numbers \n";
for ( int i =0; i <L; i++ )
{
cout << "enter value " << i << ": ";
cin >> a [ i ];
sum += a [ i ];
}
cout << "sum is: " << sum << endl;
}
```

**Example 5:-**Write C++ program, to find the minimum value in array of 8 numbers?

```
#include<iostream.h>
void main ( )
{
 int n = 8; int a [ ] = { 18, 25, 36, 44, 12, 60, 75, 89 };
int min = a [ 0 ];
for ( int i = 0; i < n; i++ )
if ( a [ i ] < min )
min = a [ i ];
cout << "The minimum number in array is: " << min;
 }
```

**Example 6:-** Write C++ program, to split the odd numbers and even numbers of one array into two arrays:

a = [ 1, 2, 3, 4, 5, 6, 7, 8, … , 20 ]
aodd = [ 1, 3, 5, 7, … , 19 ]
aeven = [ 2, 4, 6, 8, … , 20 ]

```
#include<iostream.h>
void main ( )
{
int a [ 20 ]= { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };
int aodd[20], aeven [20];
int i ,o=0, e=0;
for ( i=0 ; i<20; i++ )
if (a[i] % 2 !=0)
{
aodd[o]=a[i];
o=o+1;
}
else
{
aeven[e]=a[i];
e=e+1;
}
for ( i=0 ; i<o; i++ )
cout<<aodd[i]<<" ";
cout<<endl;
for ( i=0 ; i<e; i++ )
cout<<aeven[i]<<" ";
}
```

**Example 7:-** write C++ program to enter seven prime values for the array and finds the number of values that are greater than 50 and less than 50?

```
#include < iostream.h>
main ()
{
int i,S,L;
int a[5]={87,67,41,90,55,53,38}
for(i =0; i<=6;i++)
  {
    if(a[i]>=50)
     {
          L=L+1;
```

```
    }
   else
    {
        S=S+1;
     }
  }
  cout<<"50 اكبر من "<<L<<Endl;
  cout<<"50 اصغر من "<<S<<Endl;
}
```

**Example 8:-** write C++ program to enter ten initial values for the array and finds the number of negative values and the number of positive values?

```
#include < iostream.h>
main ()
{
int i,N,P;
int a[5]={87,-167,141,90,55,53,38,-4,-2, 1}
for(i =0; i<=9;i++)
  {
    if(a[i]>=0)
     {
          P=P+1;
     }
    else
     {
         N=N+1;
      }
  }
  cout<<"الاعداد الموجبة  "<<P<<Endl;
  cout<<"الاعداد السالبة "<<N<<Endl;
}
```

## 2. المصفوفة ذو البعد ين (Two-dimensional Array):-

يمكن للمصفوفات في + +C أن تكون متعددة الأبعاد ويمكن كذلك أن يكون كل بعد بحجم مختلف ، الاستعمال الشائع للمصفوفات متعددة الأبعاد هو تمثيل الجداول Tables التالي تحتوي على بيانات مرتبة في صورة صفوف وأعمدة ولتمثيل الجدول نحتاج لبعدين الأول يمثل الصفوف والثاني يمثل الأعمدة.

الشكل التالي يبين مصفوفة A تحتوى على ثلاثة صفوف وأربع أعمدة.

| | Column 0 | Column1 | Column2 | Column 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] |

تشبه المصفوفة ذات البعدين في طريقة تعاملها ، المصفوفة ذات البعد الواحد إلا أن لها عدادين(2 index) إحداهما عداد للصفوف ، والأخر عداد للأعمدة ويأخذ الإعلان عن المصفوفة الشكل العام التالي:

Type-specifier arraey_name [index 1][index 2];

يتم تمثيل أي عنصر في المصفوفة A على الصورة A[i][j] حيث:-
A : اسم المصفوفة.
i : رقم الصف الذي ينتمي إليه العنصر.
j : رقم العمود الذي ينتمي إليه العنصر.
لاحظ أن كل العناصر الموجودة في الصف الأول مثلاً يكون الفهرس الأول لها هو 0 وكل العناصر الموجودة في العمود الرابع يكون الفهرس الثاني لها هو 3.
يتم الإعلان عن مصفوفة a تحتوى على x صف و y عمود هكذا:

int a[x][y];

2. تهيئة عناصر مصفوفة ثنائية الأبعاد:
يمكن تمهيد قيمة المصفوفة المتعددة الأبعاد عند الإعلان عنها وذلك كالآتي:

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

حيث:

b[1][1]=4, b[1][0]=3, b[0][1]=2, b[0][0]=1
أيضاً هنا في المصفوفة متعددة الأبعاد إذا تم تمهيدها عند قيم لا يتوافق عددها مع حجم المصفوفة فإن المصرف سيملأ بقية العناصر أصفار.

**Example 1**:-write C++ program to enter the grades of 5 students into 3 subjects?

```
#include <iostream.h>
main ()
{
int m[5][3];
int I,j;
for(I=0;I<5;I++)
 {
for(j=0;j<3;j++)
  {
    cin>>m[I][j];
  }
 }
}
```

**Example 2:-**Write C++ program, to read 15 numbers, 5 numbers per row, the print them?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 5 ];
int i , j;
```

```
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 5; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 5; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
}
```

**Example 3:-** Write C++ program, to read 4*4 2D-array, then find the summation of the array elements, finally print these elements?

```
#include<iostream.h>
void main ( )
{
int a [ 4 ] [ 4 ];
int i , j, sum = 0;
for ( i = 0 ; i < 4; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 4; i++ )
for ( j = 0 ; j < 4; j++ )
sum += a [ i ] [ j ];
cout << "summation is: " << sum << endl;
for ( i = 0 ; i < 4; i++ )
{
for ( j = 0 ; j < 4; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
}
```

**Example 4:-** Write C++ program, to read 3*4 2D-array, then find the summation of each row?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 4 ];
int i , j, sum = 0;
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
{
sum = 0;
for ( j = 0 ; j < 4; j++ )
sum += a [ i ] [ j ];
```

```
cout << "summation  of row " << i << " is: " << sum << endl;
}
}
```

**Example 5:-**Write C++ program, to read 3*4 2D-array, then replace each value equal 5 with 0?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 4 ];
int i , j;
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
if ( a [ i ] [ j ] == 5 ) a [ i ] [ j ] = 0;
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 4; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
}
```

**Example 6:-**Write C++ program, to addition two 3*4 arrays?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 4 ], b [ 3 ] [ 4 ], c [ 3 ] [ 4 ];
int i , j;
cout << "enter element of array A: \n";
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
cout << "enter element of array B: \n";
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> b [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
c [ i ] [ j ] = a [ i ] [ j ] + b [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 4; j++ )
cout << c [ i ] [ j ];
cout << endl;
}
}
```

**Example 7:-**Write C++ program, to convert 2D-array into 1D-array?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 4 ];
int b [ 12 ];
int i , j, k = 0;
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
cin >> a [ i ] [ j ];
for ( i = 0 ; i < 3; i++ )
for ( j = 0 ; j < 4; j++ )
{
b [ k ] = a [ i ] [ j ];
k++;
}
for ( i = 0 ; i < k; i++ )
cout << b [ i ];
}
```

**Example 8:-**Write C++ program, print the square root of an array?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 3 ] , b [ 3 ] [ 3 ];
int i , j;
for ( i = 0 ; i < 3; i++ ) {
for ( j = 0 ; j < 3; j++ ) {
b[ i ][ j ]= sqrt(a[ i ][ j ]);
cout << b [ i ] [ j ];
}
 }
}
```

| | | |
|---|---|---|
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

**i = j**
Main diameter (diagonal)

| | | |
|---|---|---|
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

**i + j = n-1**
secondary diagonal

| | | |
|---|---|---|
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

**i > j**
under main diagonal

| | | |
|---|---|---|
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

**i < j**
above main diagonal

**Example 9:-** Write C++ program, to replace each element in the main diameter (diagonal) with zero?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 3 ];
int i , j;
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++ )
if ( i == j ) a [ i ] [ j ] = 0;
for ( i = 0 ; i < 3; i++ )
{
for ( j = 0 ; j < 3; j++ )
cout << a [ i ] [ j ];
cout << endl;
}
}
```

**Example 10:-** Write C++ program, to read 3*3 2D-array, then find the summation of the main diagonal and its secondary diagonal of the array elements, finally print these elements?

```
#include<iostream.h>
void main ( )
{
int a [ 3 ] [ 3 ];
int i , j, x , y;
for ( i = 0 ; i < 3; i++ ) {
for ( j = 0 ; j < 3; j++ ) {
cin >> a [ i ] [ j ];
if ( i == j )
x=x+a[ i ][ j ];
if ( i + j =4)
y=y+a[ i ][ j ];
}
 }
cout << "summation  of diagonal is: " << x << endl;
cout << "summation  of inverse diagonal is: " << y << endl;
}
```

**Home Work (5)**

Q1: Write a C++ program, to enter 10 initial values for the array and print the largest value between them?

Q2: Write a C++ program, to enter 10 initial values for the array, and the value of the first element A [0] changes with the last element A [9] then print the array after change?

Q3: Write C++ program, to read 3*4 2D-array, then find the summation of each col?

Q4: Write C++ program, to replace each element in the second diameter (diagonal) with zero?

Q5: Write C++ program, to replace the elements of the main diameter with the elements of the second diameter?

Q6: Write C++ program, to find the summation of odd numbers in 2D-array?

Q7:Write C++ program, to find (search) X value in 2D-array, and return the index of it's location?

Q8: Write C++ program, to convert 1D-array that size [16] to 2D-array that size of [4] [4]?

Q9: Write C++ program, to read A[ n, n ] of character, then find array B and array C, such that B contain only capital letters and C contain only small letters?

Q10: Write C++ program, to read A[ n, n ] of numbers, then put 10 instead each even positive number?

Q11: Write C++ program, to read A[ n, n ] of numbers, then put 10 instead each even positive number in the first diagonal?

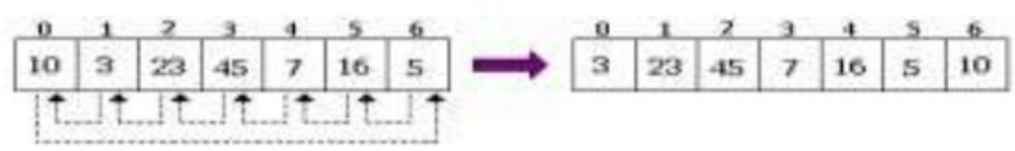
Q12: Write C++ program, to read A[ n, n ] of numbers, then find the minimum number in array?

Q13: Write C++ program, to exchange row1 and row3 in 4*3 array?

Q14: Write C++ program, to exchange row0 with col3 in 4*4 array?

Q15: Write C++ program, to find the greatest number in the second diagonal, in 3*3 array?

Q16: Write C++ program, to read X[ n ], and rotate the elements to the left by one position?



Q17: Write C++ program, to read A[ n ] and a location Z then delete the number at location Z from the array, and print the new array after deletion?

**Q**18: Write C++ program to order the array in ascending and descending order?

Q19: Write C++ program to read (n) no.s and find the average of the even no. on it?

Q20: Write C++ program to Create the array (b) from (a)?

| a | b |
|---|---|
| 1 2 3 | 6 |
| 4 5 6 | 10 |
| 7 8 9 | 10 |

Q21: Write C++ program to Create the arrays bellow?

| | |
|---|---|
| 1 1 1 1 | 2 1 1 1 |
| 2 2 2 2 | 1 2 1 1 |
| 3 3 3 3 | 1 1 2 1 |
| 4 4 4 4 | 1 1 1 2 |