

جامعة المنارة

مقرر برمجة 2

د. هلا نصار

م. أوشين داود

جامعة
المنارة

HANARA UNIVERSITY

محاضرات الأسبوع 3

الفصل الأول - 2022/2021

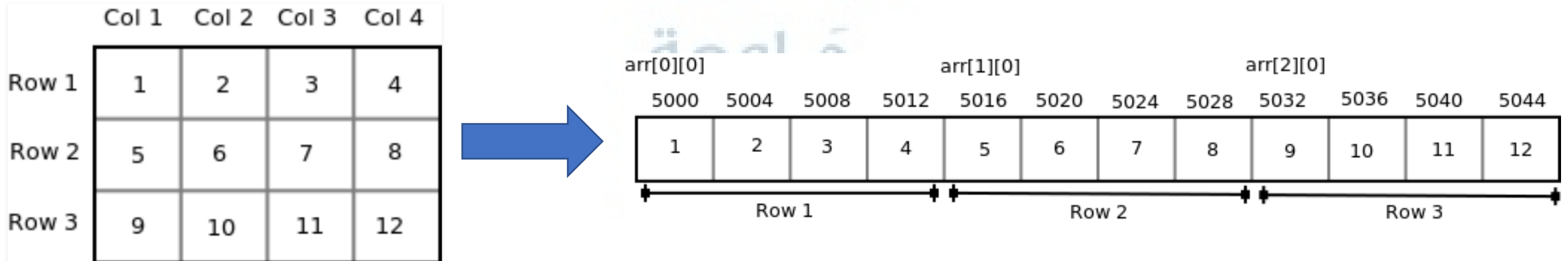
Pointers and 2-D arrays

في لغة C++ يمكن الوصول الى أي عنصر من مصفوفة ثنائية البعد بدلالة السطر والعمود كما هو موضح في المثال :

```
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
```

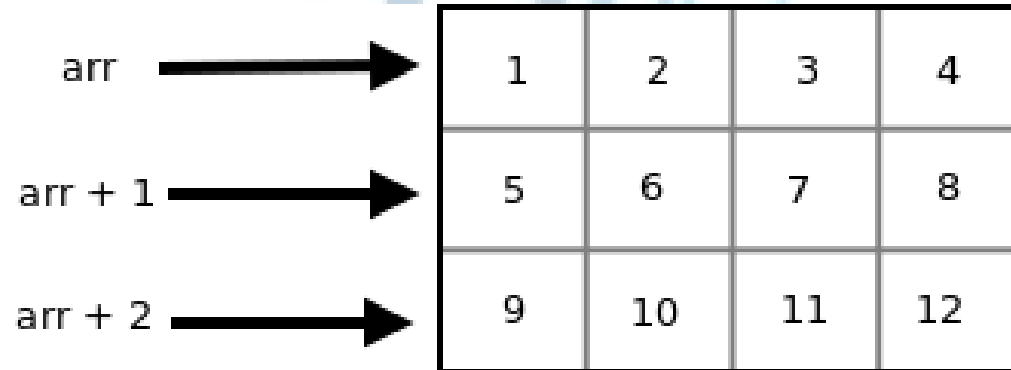
وفق الصيغة $arr[i][j]$ حيث يكون i يدل على رقم السطر و j يدل على رقم العمود.

ان مفهوم المصفوفة ثنائية البعد بالنسبة للذاكرة الفيزيائية هو مفهوم نظري، اذ يتم تخزين المصفوفة ثنائية البعد على شكل مصفوفة أحادية عناصرها الأسطر تتالي على الترتيب كما هو موضح في الشكل :



يمكن القول ان المصفوفة ثنائية البعد السابقة هي مصفوفة أحادية مكونة من 3 عناصر كل عنصر هو سطر احادي البعد مكون من أربعة اعداد صحيحة

إن اسم المصفوفة يحمل عنوان أول عنصر فيها أي هو مؤشر إلى أول عنصر منها أي عنوان أول عنصر من السطر الأول. عندما نزيد المؤشر `arr` بمقدار 1 أو 2 فهذا يعني أنه يشير إلى العنصرين الثاني والثالث من المصفوفة أي إلى السطرين الثاني والثالث على التوالي. ولكن يبقى السؤال كيف سيتم الوصول إلى العنصر الموجود في كل سطر من الأسطر؟



الوصول الى عناصر المصفوفة ثنائية البعد باستخدام اسم المصفوفة كمؤشر الى كل سطر على حدى

```
#include <iostream>
using namespace std;
int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
for (int i=0; i<3;i++)
    cout <<(arr + i)<< endl;
for (int i=0; i<3;i++)
    cout << * (arr + i)<< endl;
return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
for (int i=0; i<3;i++)
    cout <<*(*(arr + i))<< endl;
return 0;
}
```

الوصول الى عناصر المصفوفة ثنائية البعد باستخدام اسم المصفوفة كمؤشر الى كل سطر على حدة

```
#include <iostream>
using namespace std;
int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
for (int i=0; i<3;i++)
    for (int j=0 ; j<4; j++)
        cout << *(arr + i) + j<< endl;
return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
for (int i=0; i<3;i++)
    for (int j=0 ; j<4; j++)
        cout << (*(arr + i) + j)<< endl;
return 0;
}
```

HANARA UNIVERSITY

أعد حل المثال السابق بالاعتماد على المؤشر الى مصفوفة سطرية

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

int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
int (*ptr)[4];
ptr = arr;
for (int i=0; i<3;i++)
    for (int j=0 ; j<4; j++)
        cout<< *((ptr + i) + j)<< endl;
return 0;
}
```

أعد حل المثال السابق بالاعتماد على المؤشر الى مصفوفة سطرية

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

int main()
{
int arr[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
int (*ptr)[4];
ptr = arr;
for (int i=0; i<3;i++)
    for (int j=0 ; j<4; j++)
        cout<< *(ptr[i]+j)<<endl;
return 0;
}
```

الوصول الى عناصر المصفوفة 2D من خلال تعريف مؤشر الى اول عنصر منها

```
#include <iostream>
using namespace std;
int main()
{
int arr[3][4] = {{1, 2, 3, 4},{5, 6, 7, 8},{9, 10, 11, 12}};
// pointer ptr pointing at array arr
int *ptr = &arr[0][0];
// other variables
int ROWS = 3, COLS = 4, TOTAL_CELLS = ROWS * COLS;
// print the elements of the array arr via pointer ptr
for (int i = 0; i < TOTAL_CELLS; i++)
    cout << *(ptr + i)<<endl;
return 0;
}
```


Array Of Pointers

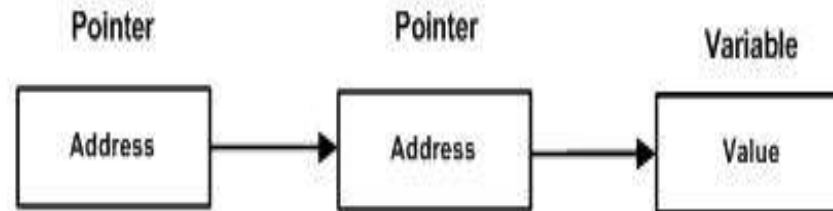
```
#include <iostream>
using namespace std;

int main()
{int myarray[5] = {2,4,6,8,10};
int *ptr[5]; //array of pointers
for(int i=0;i<5;i++)
    {ptr[i] = &myarray[i];}
for (int i = 0; i < 5; i++)
    {cout << "Value of myarray[" << i << "] = ";
    cout << *ptr[i] << endl; }
return 0;
}
```

Pointers to pointers

في لغة C++ يمكن تعريف مؤشر الى مؤشر أي نعرف متحول من نوع مؤشر يحمل عنوان مؤشر آخر يؤول الى قيمة ما لكن يجب أن يكون المؤشر والمؤشر لمؤشر من نفس نوع المعطيات التي يشير إليها.

يتم تعريف مؤشر الى مؤشر وفق الصيغة التالية :
`int **var;`



مؤشر الى مؤشر كما هو موضح في المثال التالي

```
#include <iostream>
using namespace std;
int main ()
{int var;
int *ptr;
int **pptr;
var = 22;
// take the address of var
ptr = &var;
// take the address of ptr using address of operator &
pptr = &ptr;
// take the value using pptr
cout << "Value of var :" << var <<" &var= " <<&var<< endl;
cout << "Value available at *ptr :" << *ptr << " ptr= " << ptr<< " &ptr =
"<<&ptr<<endl;
cout << "Value available at **pptr :" << **pptr <<" pptr= " << pptr<< endl;
return 0;
}
```

QUIZ

عرف مصفوفة ثنائية البعد مربعة بعدها 5 وهيئةا بقيم تختارها واطبع عناوين عناصر السطر الثاني

```
#include <iostream>
using namespace std;
int main()
{
int arr[5][5] = { {1, 2, 3, 4,66}, {5, 6, 7, 8,77}, {9, 10, 11, 12,99},{11,22,33,44,55},{10,20,30,40,50} };
for (int j=0 ; j<5; j++)
cout << *(arr + 1) + j<< " " << (*(arr + 1) + j)<<endl;

return 0;
}
```

انتهت تمارين الأسبوع 3

جامعة
المنارة
HANARA UNIVERSITY