

Second class

Subject : Data Structures

Lecturer : Israa Mahmood Hayder

Lecture no.5

هيكل البيانات المركبة (Compound Data Structures) - الاسبوع - الخامس -

B// Rationale (مبررات الوحدة) :-

Multi dimensional Arrays are a very basic data structure representing a group of similar elements, accessed by index. Array data structure can be effectively stored inside the computer and provides fast access to the all its elements. Let us see an advantages and drawbacks of the arrays.

C// Central Ideas (الفكرة المركزية):-

- Declaring a Multidimensional Array
- Representation of two dimensional arrays using:
 - 1) Row major method
 - 2) Column major method

D// Objectives (أهداف الوحدة):-

After studying this unit, the student will be able to:-

- Define Two Dimensional array data Structure
- Learn how to declare and use the arrays in C++

- Represent the two dimensional array and calculate the memory Location of any element using:
 - 1- Row major method
 - 2- Column major method

Declaring an Array

The way to declare an array depends on the programming language used to write your program. In Java, there are two techniques for declaring an array. You can declare and initialize an array either where memory is allocated at compile time or where memory is dynamically allocated at runtime. *Allocation* is another way of saying reserving memory.

Let's begin by declaring an array where memory is reserved when you compile your program. This technique is similar in Java, C, and C++, except in Java you must initialize the array when the array is declared. There are four components of a statement that declares an array. These components are a data type, an array name, the total number of array element to create, and a semicolon (;). The semicolon tells the computer that the preceding is a statement. Here's the declaration statement in C and C++:

```
int grades[10];
```

In Java, you must initialize the array when the array is declared as shown here. The size of the array is automatically determined by counting the number of values within the braces. Therefore, there isn't any need to place the size of the array within the square brackets:

```
int[] grades = {0,0,0,0,0,0,0,0,0,0};
```

The *data type* is a keyword that tells the computer the amount of memory to reserve for each element of the array. In this example, the computer is told to reserve enough memory to store an integer for each array element.

The *array name* is the name you use within a program to reference an array element. The array name in this example is grades. The number within the square brackets is the total number of elements that will be in the array. The previous statements tell the computer to create an array of 10 elements.

Declaring a Multidimensional Array

A multidimensional array is declared similar to the way you declare a one-dimensional array except you specify the number of elements in both dimensions. For example, the multidimensional array declared as follows in C or C++:

```
int grades[3][4];
```

You assign a value to an element of a multidimensional array with an assignment statement similar to the assignment statement that assigns a value to a single-dimensional array, as shown here:

```
grades[0][0] = 1001;
```

You must specify the index for both dimensions. In this example, the integer 1001, which is a student ID, is assigned to the first element of the first set of elements in the grades array.

Storage structure for: Two Dimensional Arrays

In this type the data will be stored row by row (row-major order) or column by column (column-major order)/column wise method.

1- Storing in row-major order:

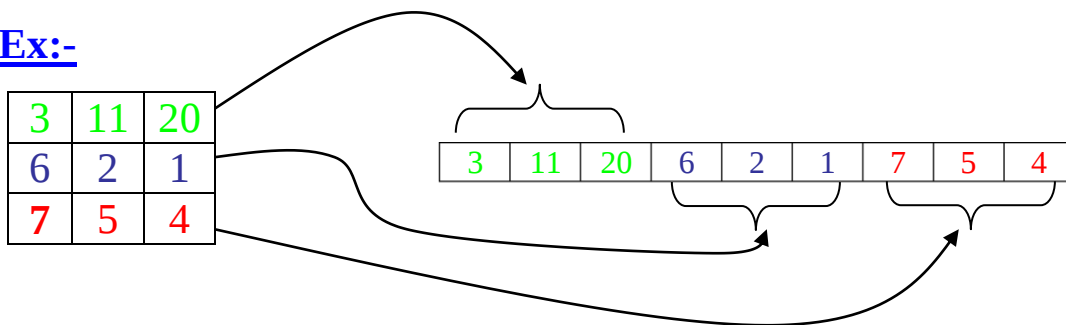
$$\text{Loc}(A[i,j]) = \text{Loc} + C * (M * (I-1) + (J-1))$$

Such that

Loc is the starting address to store

M is total no of Columns.

Ex:-



2- Storing in column-major order:-

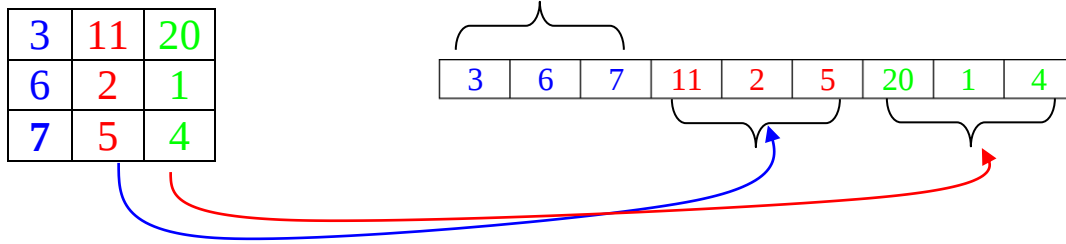
$$\text{Loc}(A[i,j]) = \text{Loc} + C * (N * (J-1) + (I-1))$$

Such that

N is total no of Rows.



Ex:-



Ex1:- Find the address of the element B(2,3) in memory of the array B(5,6). Assume 2 byte storage and the starting address to store the array is 500 assuming:-

- 1- row-major storage.
- 2- Column-major storage.

Sol:-

$$1) \text{ Loc (B[2,3])} = 500 + 2 (6 * (2 - 1) + (3 - 1)) = 516$$

$$2) \text{ Loc (B[2,3])} = 500 + 2 (5 * (3 - 1) + (2 - 1)) = 522$$

Ex2:- If you have the following data :-

6	200	91
10	450	97
5	230	98

If you know the starting address is 1224 and 2 byte storage find the location of the elements 450 and 230 in the memory. Give storage mapping into memory assuming column-major storage .

Sol:-

$$1- \text{ Loc (X[2,2])} = 1224 + 2 (3 * (2 - 1) + (2 - 1)) = 1238$$

$$2- \text{ Loc (X[3,2])} = 1224 + 2 (3 * (2 - 1) + (3 - 1)) = 1234$$

Arrays in C++: (Multi dimensional arrays):

```
int num[4][5];
      ↑  ↑
      rows columns
```

Ex:

```
# include <iostream.h>
Int main( )
{
    int nu[4][5] , total=4 , i,j ;
    for (i=0 , i<4 ; i++)
        for ( j=0 , j<5 , j++)
            { nu[i][j] = i+j ;
              total = total+nu[i][j] ;
            }
    for (i=0 , i<4 ; i++)
        { for ( j=0 , j<5 , j++)
            cout << nu[i][j]<<"\t";
            cout <<"\n";
        }
    cout<<"total="<<total;
    return 0 ;
}
```

قراءة المصفوفة وإيجاد مجموع عناصرها

طبع عناصر المصفوفة

Quiz1: Write program to read array a[3x4] if integers and find the sum of first row and last column.

Quiz2

Assume that 2 bytes of storage are used to store each element of the array A(3×2) at the starting address 800. Find the address of the element A(3,1) using:

- 1- Row-Major method
- 2- Column-Major method

References:

1 Data Structures Demystified, by Jim Keogh and Ken Davidson, ISBN:0072253592, McGraw-Hill/Osborne © 2004.

الجزء الاول، الطبعة الاولى، جون ر. هيوبارد، الدار الدولية سلسلة ملخصات شوم للبرمجة، -1

بلغة C++ للاستثمارات الثقافية، ٢٠٠٠

3- هياكل البيانات / الطبعة الثانية، تاليف د. عصام الصفار، اصدارات السفير للنشر/ بغداد، ٢٠٠١

٤- الحقيبة التعليمية مادة "هياكل البيانات"، اعداد : نفارت الياس يوسف، المعهد التقني كركوك