ACADEMIC YEAR 2022/2023

[KK] DATA STRUCTURE

LECTURE NOTES AND ASSIGNMENT

ROBERTUS HUDI
FACULTY OF COMPUTER SCIENCE, INFORMATICS DEPARTMENT
Universitas Pelita Harapan

CHAPTER IV – 2D ARRAY

LECTURE NOTES

Overview

- Multi-dimensional arrays are declared by providing more than one set of square [ ] brackets after the variable name in the declaration statement.
- One dimensional arrays do not require the dimension to be given if the array is to be completely initialized. By analogy, multi-dimensional arrays do not require **the first** dimension to be given if the array is to be completely initialized. All dimensions after the first must be given in any case.
- For two dimensional arrays, the first dimension is commonly considered to be the number of rows, and the second dimension the number of columns. We will use this convention when discussing two dimensional arrays.
- Two dimensional arrays are considered by C/C++ to be an array of (single dimensional arrays). For example, "`int numbers[5][6]`" would refer to a single dimensional array of 5 elements, wherein each element is a single dimensional array of 6 integers. By extension, "`int numbers[12][5][6]`" would refer to an array of twelve elements, each of which is a two dimensional array, and so on.
- Another way of looking at this is that C stores two dimensional arrays by rows, with all elements of a row being stored together as a single unit. Knowing this can sometimes lead to more efficient programs.

	Column 0	Column 1	Column 2	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]

- Multidimensional arrays may be completely initialized by listing all data elements within a single pair of curly {} braces, as with single dimensional arrays.

```
#include <bits/stdc++.h>
using namespace std;

int main(void)
{
    int i,j;
    int a[3][2] = {1, 4, 5, 2, 6, 5};
    for(i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

- It is better programming practice to enclose each row within a separate subset of curly {} braces, to make the program more readable. This is required if any row other than the last is to be partially initialized. When subsets of braces are used, the last item within braces is not followed by a comma, but the subsets are themselves separated by commas.
- Multidimensional arrays may be partially initialized by not providing complete initialization data. Individual rows of a multidimensional array may be partially initialized, provided that subset braces are used.
- Individual data items in a multidimensional array are accessed by fully qualifying an array element. Alternatively, a smaller dimensional array may be accessed by partially qualifying the array name. For example, if "`data`" has been declared as a three dimensional array of floats, then `data[1][2][5]` would refer to a float, `data[1][2]` would refer to a one-dimensional array of floats, and `data[1]` would refer to a two-dimensional array of floats. The reasons for this and the incentive to do this relate to memory-management issues that are beyond the scope of these notes.

Reading Elements of 2D Array

Using `scanf`, a single individual element of 2D Array could be read this way:

```
for(row=0;row<3;row++)
{
    for(col=0;col<2;col++)
    {
        scanf("%d ", &a[row][col]);
    }
}
```

Passing Array to Function

- Recall that we know two methods for passing ordinary data to functions:
  - Pass-by-Value, in which the function receives a copy, and all changes are local, and
  - Pass-by-Reference, in which the names of the variables in the called and calling functions become aliases, and changes made in the called function DO affect the variables in the calling function.
- If an individual element of an array is passed to a function, it is passed according to its underlying data type. So, if `nums` was declared as a one-dimensional array of `int`s, then passing `nums[i]` to a function would behave the exact way as passing any other `int` - Either pass-by-value or pass-by-reference, depending on how the function is written.
- When an entire array is passed to a function, however, it is always passed by reference.
  - (It is passed by pointer/address, which is not covered here, but functionally it's by-reference)
  - The net result is that when an entire array is passed to a function, and the function changes variables stored in that array, it does affect the data in the calling function's array.
  - Because arrays are passed by reference, there is generally no need for a function to "return" an array. - It merely needs to fill in an array provided by the calling function. (It is possible for functions to return arrays but it requires the use of pointers and addresses, and frequently dynamic memory allocation, all of which is beyond the scope of this course.)
  - To prevent the function from changing the array values, the array parameter can be modified with the keyword const.

- When an entire array is passed to a function, the size of the array is usually passed as an additional argument.
- For a one-dimensional array, the function's formal parameter list does not need to specify the dimension of the array. If such a dimension is provided, the compiler will ignore it.
- For a multi-dimensional array, all dimensions except the first must be provided in a function's formal parameter list. The first dimension is optional and will be ignored by the compiler.
- A partially qualified multi-dimensional array may be passed to a function and will be treated as an array of lower dimension. For example, a single row of a two-dimensional array may be passed to a function which is expecting a one-dimensional array.

ASSIGNMENT

Problem Set: Land Valuation System

Problem Definition:

Dimitri is a businessman who loves investing in land properties. He always has a certain way to evaluate a land (L) into a N×M rectangular land area and put a profit value of each block ($L_{i,j}$). He then will purchase some parts of the land or all the land if the profit could be maximized. **However, Dimitri always follows his principal to always purchase a land in a complete rectangular block(s)**. Assume this rectangular shape area with a marked profit value for each block:

6	-5	-7	4	-4
-9	3	-6	5	2
-10	4	7	-6	3
-8	9	-3	3	-7

Based on the land mapping above, the land is mapped into 4×5 blocks where Dimitri calculates the most profit, he could get is to purchase a land with following blocks:

6	-5	-7	4	-4
-9	3	-6	5	2
-10	4	7	-6	3
-8	9	-3	3	-7

Resulting with 17 profit points, he chooses 4×5 blocks rectangular that has value of {4, 7, 9, -3}. There is one extra condition, if it is impossible for Dimitri to get any profit, tell him to not buy any block in that land area. While he has his own method to evaluate land profit of each block, he still need help to maximize his profit. Dimitri is now hiring you to calculate the maximum profit he could obtain. Let's work!

Restrictions:

First line of **input** consists of an integer T ($1 \le T \le 50$) represents the number of test cases.

First line of test case consists of N and M, represent the size of the field. ($3 \le N, M \le 100$)

Next N lines consist of M integers denoting $L_{i,j}$ the value of the area in that location. ($-10^6 \le L_{i,j} \le 10^6$).

For each case of **output**, print "Case #X: " where X is the test case number starts from 1, followed by the profit Dimitri could get, but if it is impossible to get any profit, print "NO"**.**

Sample Input:

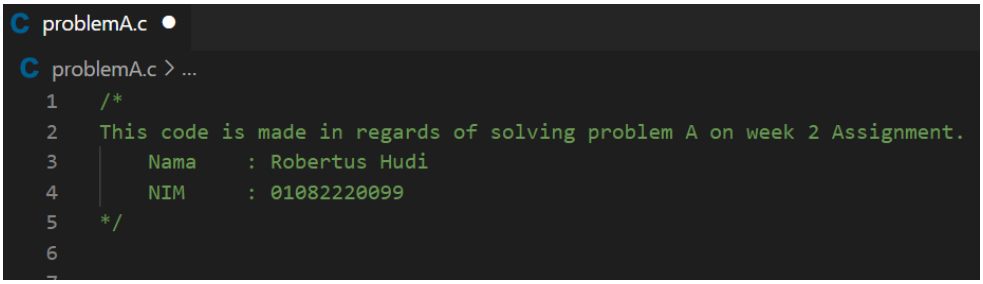
```
3
4 5
6 -5 -7 4 -4
-9 3 -6 5 2
-10 4 7 -6 3
-8 9 -3 3 -7
5 5
-5 -6 3 1 0
9 7 8 3 7
-6 -2 -1 2 -4
-7 5 5 2 -6
3 2 9 -5 1
3 3
-1 -1 -1
-1 -1 -1
-1 -1 -1
```

Sample Output

```
Case #1: 17
Case #2: 35
Case #3: NO
```

Instructions

- Submission file must be 1 compressed zip or rar file containing 1 file of source code and 1 file of code explanation in word document.
- The compressed file should be named as follows: Ch2_Assignment_<NAME>.zip or Ch2_Assignment_<NAME>.rar; Sample: Ch2_Assignment_Robertus Hudi.zip
- Source code should be named in regard of the problem numbering mentioned above. Sample: problemA.c or problemA.cpp or problemA.java
- For each section of the code, you have to put a following comment line(s) before the code begins. Sample:



- Submission slot is available on the learn.uph.edu.