

Two-dimensional Arrays

- **For multiple-choice and essay questions:** Present your work to the PDF file whose name is <StudentID>_Writing.pdf, where <StudentID> is your student number.
 - **Multiple-choice questions:** Pick the most probable choice and explain for each choice why you choose/do not choose it. There are sometimes multiple correct choices.
 - **Essay questions:** Give a comprehensive answer that covers all aspects of the problem
- **For programming question:** Write your source codes in C++/Python 3. Do not use any additional libraries/open source codes without permission.
 - **C++ programming language:** Name your Visual Studio solution such that the execution filename is <MSSV>_<Bi>.exe, where StudentID is your student number, B denotes programming questions, and i is the question index (e.g., B1, B2, etc.). Please submit the whole solution folder (after delete all bulky intermedia files). Do not include the execution file in your submission to avoid spreading dangerous virus.
 - **Python programming language:** name your Python files in a similar way to what described above
 - Programming questions sometimes require brief explanation. Present your work to the PDF file described above.
- TA supports solutions for multiple-choice and essay questions after the submission deadlines.
- Students can contact TA through email (Ms. Nguyễn Ngọc Thảo: nnthao@fit...).

A. Multiple-choice questions and essay questions (10 marks)

A.1. Given a two-dimensional array as follows.

```
int a[3][4] = { {0, 1, 2, 3},  
                {4, 5, 6, 7},  
                {8, 9, 10, 11} };
```

Which of the following statements can be used to access the seventh elements? (There may multiple correct options)

- a. `a[0][7]`
- b. `a[0][6]`
- c. `a[1][2]`
- d. `a[7][0]`
- e. `a[6][0]`
- f. None of the above

A.2. Which of the following declarations is VALID?

- a. `char arr[][] = {'a', 'b'};`
- b. `char arr[2][2] = {{'a', 'b'}, {'c', 'd'}};`
- c. `char arr[2][] = {{'a', 'b'}, {'c', 'd'}};`
- d. `char arr[][] = {{'a', 'b'}, {'c', 'd'}};`

A.3. If you want to pass a multi-dimensional array (i.e. the number of dimensions is greater than 1, in general, e.g. `a[4][3][2]`), which dimension(s) requires its size to be specified?

- a. All dimensions, e.g. `void func(int a[4][3][2]);`
- b. All dimensions except the last dimension, e.g. `void func(int a[4][3][]);`
- c. All dimensions except the first dimensions, e.g. `void func(int a[][3][2]);`
- d. No dimension, e.g. `void func(int a[][][]);`

A.4. Given a two-dimensional array as follows.

```
double a[3][4] = { {1.2, 9.0, 3.2},  
                   {9.2, 0.5, 1.5, -1.2},  
                   {7.3, 7.9, 4.8} } ;
```

What is the value of the element `a[3][0]`?

- a. 7.3
- b. 9.2
- c. 7.9
- d. 0
- e. Undetermined

A.5. Assume that you need to create a two-dimensional array from the matrix shown aside. Note that there is an empty cell at row 2 and column 3.

12	-9	8
7	14	
-32	-1	0

Which of the following array declarations is VALID?

- a. `double table[][3] =`
`{ 12, -9, 8,`
`7, 14,`
`-32, -1, 0} ;`
- b. `double table[][3] =`
`{ {12, -9, 8}`
`{7, 14}`
`{-32, -1, 0} };`
- c. `double table[][3] =`
`{ {12, -9, 8},`
`{7, 14, 0},`
`-32, -1, 0} };`
- d. `double table[][3] =`
`{ {12, -9, 8},`
`{7, 14},`
`{-32, -1, 0} };`

A.6. Is it possible to use a one-dimensional array to store the elements of a two-dimensional array, and vice versa? If yes, give an example for each case and explain why it is convenient to have both types of arrays? If no, briefly give your reasons.

A.7. What is the output of the following program? Explain why.

```
int m2(int a[], int n){
    int v = a[0];
    for (int i = 1; i < n; i++)
        if (v < a[i])
            v = a[i];
    return v;
}
int main(){
    int a[2][4] = { { 3, 4, 5, 1 }, { 33, 6, 1, 2 } };
    for (int row = 0; row < 2; row++)
        cout << m2(a[row], 4);
    return 0;
}
```

A.8. The following code segment aims to print the content of a two-dimensional array. Identify all possible errors in the following code segment and fix them.

```
int main(){
    int a[][] = { {1, 2, 3}, {4, 5, 6}};
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            cout << a[j][i];
    return 0;
}
```

A.9. What is the output of the following program? Explain why.

```
void main(){
    int a[][2][2] = { { { 1, 2 }, { 3, 4 } }, { { 5, 6 }, { 7, 8 } } };
    cout << a[1][0][0]);
}
```

A.10. What is the output of the following program? Explain why.

```
int m3(int a[][2]){
    int v = a[0][0];
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 2; j++)
            if (v < a[i][j])
                v = a[i][j];
    return v;
}

void main(){
    int a[][2][2] = { { { 1, 2 }, { 3, 4 } },
                      { { 5, 6 }, { 7, 8 } } };
    cout << m3(a[0]);
}
```

B. Programming questions (10 marks)

Students do not have to check the validity of the input.

- B.1.** Write a program that receives a matrix as input and converts it to an one-dimensional array.
Write a program that receives an one-dimensional array and converts it to a matrix.
- B.2.** Write a program that receives a matrix as input and performs the following operations.
- Check whether it is a square matrix, a symmetric matrix or an identity matrix. Note the these types of matrix are not mutually exclusive.
 - Print all elements on the main diagonal and the counter diagonal if this is a square matrix
 - Find the minimum and the maximum elements of the matrix.
 - Find the median of all elements of the matrix.
 - Determine the transpose of the matrix
- B.3.** Write a program that receives an input **square** matrix and then calculates the trace and determinant of the matrix
- B.4.** Write a program that receives two matrices as input. If these matrices have the same size, the program calculates their element-wise product. If the number of columns of the left matrix is the same as the number of rows of the right matrix, the program calculates their common matrix product.
- B.5.** Write a program that creates a matrix whose content as follows. Neither array initialization nor accessing all 25 elements is allowed.

0	1	1	1	1
-1	0	1	1	1
-1	-1	0	1	1
-1	-1	-1	0	1
-1	-1	-1	-1	0

- B.6.** Write a program that receives an input **square** matrix and then rotates the matrix 90° from left to right.

For example, from	1 2 3	to	7 4 1
	4 5 6		8 5 2
	7 8 9		9 6 3