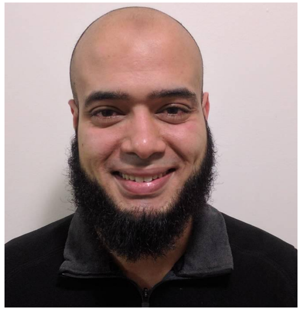
C++ Programming

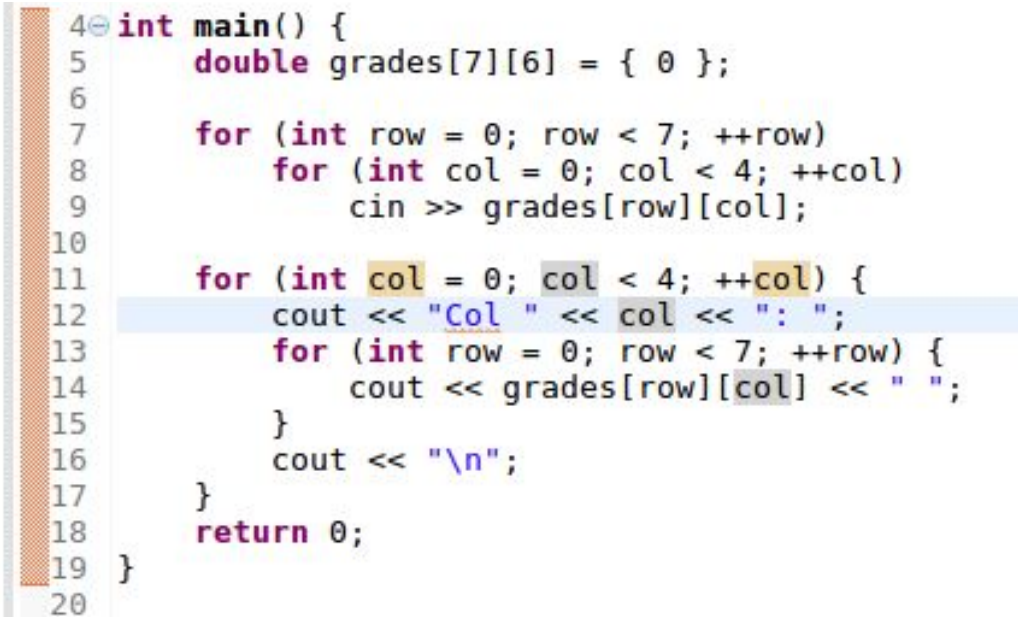
Multidimensional Arrays 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

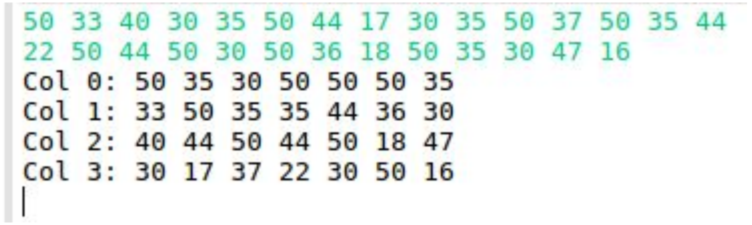


Column Row Order



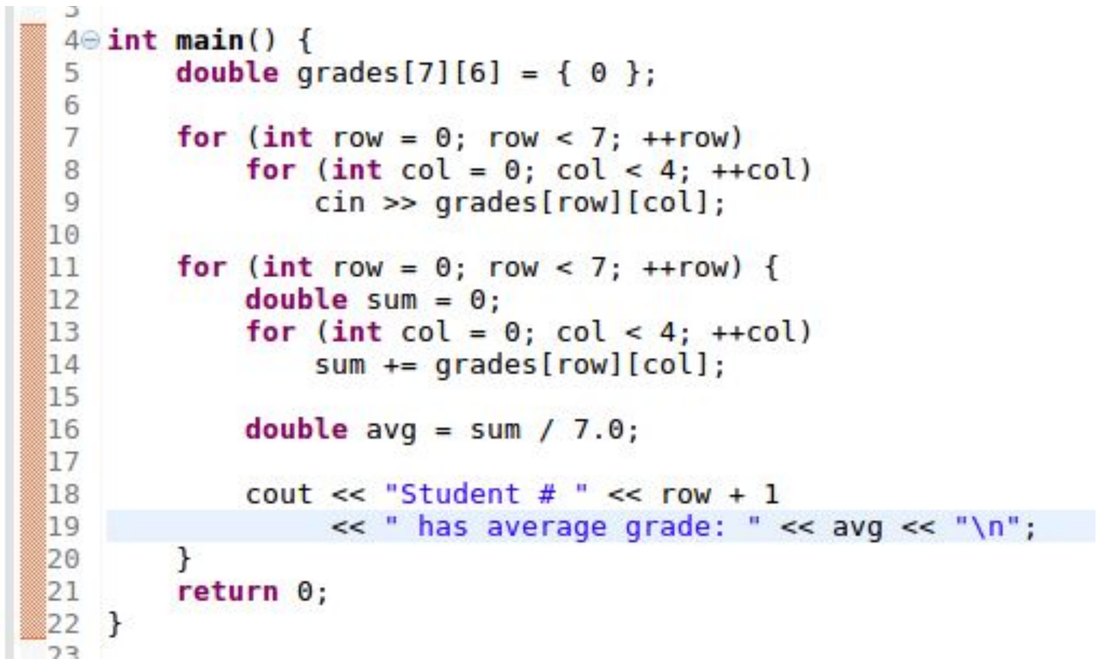
```
int main() {
    double grades[7][6] = { 0 };

    for (int row = 0; row < 7; ++row)
        for (int col = 0; col < 4; ++col)
            cin >> grades[row][col];

    for (int col = 0; col < 4; ++col) {
        cout << "Col " << col << ": ";
        for (int row = 0; row < 7; ++row) {
            cout << grades[row][col] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

- We can also see it from the columns perspective
    - Note: This is slower :)

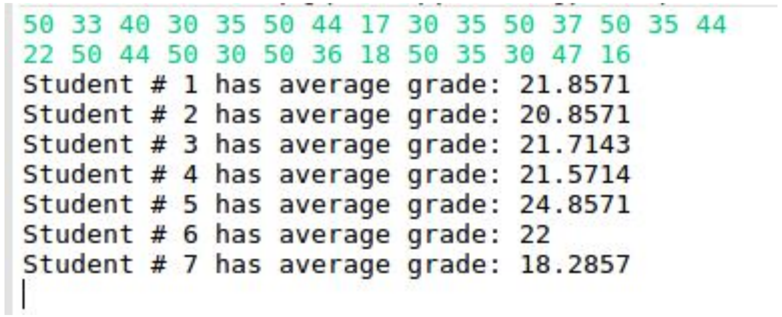


```
50 33 40 30 35 50 44 17 30 35 50 37 50 35 44
22 50 44 50 30 50 36 18 50 35 30 47 16
Col 0: 50 35 30 50 50 50 35
Col 1: 33 50 35 35 44 36 30
Col 2: 40 44 50 44 50 18 47
Col 3: 30 17 37 22 30 50 16
```

Let’s compute average grade per student



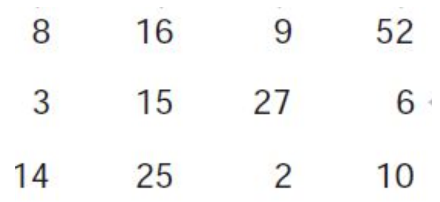
```cpp
int main() {
    double grades[7][6] = { 0 };

    for (int row = 0; row < 7; ++row)
        for (int col = 0; col < 4; ++col)
            cin >> grades[row][col];

    for (int row = 0; row < 7; ++row) {
        double sum = 0;
        for (int col = 0; col < 4; ++col)
            sum += grades[row][col];

        double avg = sum / 7.0;

        cout << "Student # " << row + 1
             << " has average grade: " << avg << "\n";
    }
    return 0;
}
```



```
50 33 40 30 35 50 44 17 30 35 50 37 50 35 44
22 50 44 50 30 50 36 18 50 35 30 47 16
Student # 1 has average grade: 21.8571
Student # 2 has average grade: 20.8571
Student # 3 has average grade: 21.7143
Student # 4 has average grade: 21.5714
Student # 5 has average grade: 24.8571
Student # 6 has average grade: 22
Student # 7 has average grade: 18.2857
```

Flatten an array

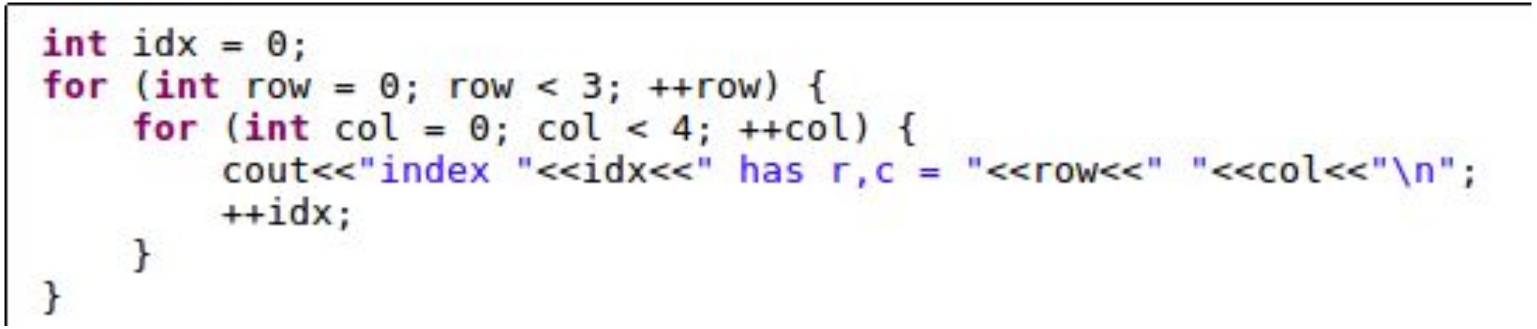
- To flatten array, means convert to 1D array
- You simply put values from rows in order
- E.g. array 1D now is:
  - **8 16 9 52** 3 15 27 6 **14 25 2 10**



8	16	9	52
3	15	27	6
14	25	2	10

Flatten an array

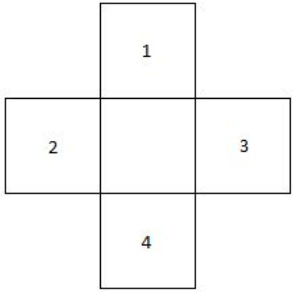
- Let say the 2D array is 3x4. Then new 1D array has length 12 also
  - If we have position (i, j) in 2D array, what is index in 1D array?
  - If we have index in 1D array, what is the position (i, j) in 2D array?
  - **Find a simple formula** for each of them. Use the following code to enumerate

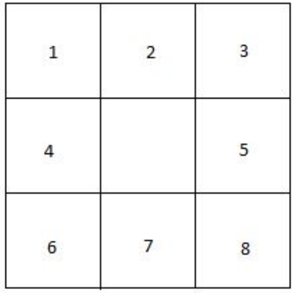


```
int idx = 0;
for (int row = 0; row < 3; ++row) {
    for (int col = 0; col < 4; ++col) {
        cout<<"index "<<idx<<" has r,c = "<<row<<" "<<col<<"\n";
        ++idx;
    }
}
```

Position neighbours

- For a position (i, j)
  - Sometimes we use 4 neighbours
    - **up, right, down, left**
  - Sometimes we use 8 neighbours
    - **up, right, down, left**, up right, up left, down right, down left
    - Given (i, j), can u use a loop of 8 steps and print theses 8 positions, elegantly?





Multidimensional Arrays

- What if we have 5 years. For each year, we have 100 students and 20 subjects? How to represent?
  - 5 Arrays, each one is 2D array [100][20]
  - Not convenient
- C++: double grades[5][100][20];
  - 3D array
  - grades[2][70][8];
  - Grade for the 3rd year, student #71, 9th subject
  - This is 2 * 70 * 8 double numbers
- You can do bigger arrays
  - Int results[10][10][10][10][10][10];
    - This is 1000,000 numbers. Be careful.

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."