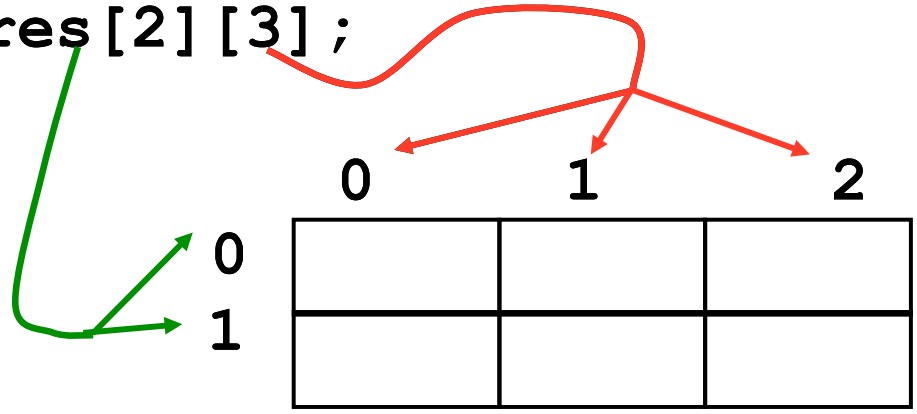
Arrays (Continued)

Two dimensional arrays

- A grid of data!

```
int testScores[2][3];
```




```
testScores[0][0] = 99;
testScores[0][1] = 80;
testScores[0][2] = 88;
testScores[1][0] = 89;
testScores[1][1] = 77;
testScores[1][2] = 85;
```

Why use 2D arrays?

- Hold the scores for each student in one array.

```
const int BOB = 0;
const int ALICE = 1;
const int MIDTERM1 = 0;
const int MIDTERM2 = 1;
const int FINAL = 2;
int testScores[2][3] = { {0, 0, 0},
                         {0, 0, 0} };
testScores[BOB][MIDTERM1] = 99;
testScores[ALICE][FINAL] = 85;
```

- Which values are we setting above?
- How do we set Alice's Midterm2 score?
- What is stored in **`testScores[0][1]`** ?

Practice

- Use a two dimensional array to store the scores of 8 Pacific Volleyball games. Store the opponent names in a separate one dimensional array. Read these values from PV.txt. Pacific's score is listed first

```
Concordia 3 2
Schreiner 3 0
Wartburg 3 2
Iowa 3 2
LaVerne 3 2
UCSC 2 3
CalLutheran 0 3
Pomona 2 3
```

- Print the name of the first team that Pacific beat
- Print the name of the last team that Pacific beat
- Print the name of the first team that beat Pacific

Practice

- Using the array below, calculate:
  - the average score on each assignment
  - the average score for each student
  - assume the array already contains data

```
const int NUMOFSTUDENTS = 24;
const int NUMOFASSIGNMENTS = 6;

int testScores[NUMOFSTUDENTS ][NUMOFASSIGNMENTS ];
```

N-Dimensional Arrays

```
// cost of seats in a theatre
//
// 4 sections, each section has
// 20 rows with 30 seats each.

double seats[4][20][30];

seats[0][0][0] = 100.00;
seats[2][0][0] = seats[1][0][0] / 2;
seats[3][19][25] = 10.00;

// we can have as many dimensions as
// necessary in an array
```