Class 38: Multi-dimensional arrays

Held: Tuesday, 12 April 2011

Summary: We consider multi-dimensional arrays. That is, we consider arrays of arrays. We examine how one declares them, why one might use them, and some complexities in their use.

Related Pages:
- EBoard.
- Lab: Multi-Dimensional Arrays.
- Reading: K&R 5.6-5.9.

Notes:
- Some answer to questions on multi-dimensional arrays posted.
- Missing questions from DG, MH, KI, JTL.
- Really late: AH, JJ.
- No reading for Wednesday!
- I will reserve time at the start of class for questions on on assignment 7.
- More on preregistration: Compilers, Discrete Math, etc.
- EC for either of Tuesday’s Town-Hall Meetings on the Forum [academic]
- EC for Thursday’s Convocation [academic]
- EC for Thursday’s CS Extra [academic]
- EC for Friday’s CS Table - Wolfram TED Talk, [academic]
- EC for baseball (2 games Saturday, 2 games Sunday) [peer]
- EC for Bob’s Benefit Bash Friday [peer]

Overview:
- Multi-dimensional arrays.
- Why?
- Arrays of pointers vs. Arrays of arrays.
- Lab.

Multi-Dimensional Arrays

- You can make arrays of anything.
- Even arrays of pointers and arrays of other arrays.
- What’s the syntax for declaring?
- An array of pointers
TYPE *NAME[SIZE];

- An array of arrays

    TYPE NAME[ROWS][COLS];

- You can initialize them when declaring.

    int stuff[2][3] = { { 1, 2, 3 },
                        { 4, 5, 6 } };

**Why Do We Need Multi-Dimensional Arrays?**

- When we have arrays of strings.
- When we want to represent matrices for computation (e.g., for those physics types)
- When we want to represent compound data, one way to represent compound data is an array. (E.g., we might represent a student in this class by a five element array of strings: Last name, First name, Student ID, Major, and Grade.)
  - A collection of compound data can then be represented as an array of arrays. (Or, more precisely, an array of arrays of arrays.)
- As you’ve no doubt determined, tables of numbers are often useful. Two dimensional arrays simulate tables.

**Arrays vs Pointers, Revisited**

- We’ve seen that arrays and pointers are similar in some ways, but different in others.
- What’s the difference between an array of pointers and an array of arrays?

**Lab**

- Start The lab.