CS 1132 lecture 7

I. Boundary conditions
    a. Padding may not be ideal in practice for large matrices – requires copying lots of memory. But still a good illustration of slicing and concatenation techniques
    b. Vectorized code may be harder to read, especially for those not used to it
        i. If serial code is easier to think about, start with that
    c. Alternative to padding: use variable slice bounds
        i. Functions min(), max() are useful for clamping to bounds

II. Example: vectorized Monte Carlo simulation

III. Vectorized relational and logical operations
    a. With vector operands, yield logical ("Boolean") vector results
    b. Don't double &, | for logical operations (no short-circuiting)

IV. Advanced slicing
    a. Can slice with a permutation vector of indices
    b. Can slice with a logical vector of equal length
    c. Much more concise than for-loops
    d. Example: select students who improved on second exam

V. Logical vectors
    a. Can count how many are true with sum()
    b. Can invert selection with ~ (in index)
    c. Can convert to vector of indices with find()

VI. rollDie revisited
    a. Sum of two uniform distributions is not a uniform distribution
    b. If multiple samples are required, must make multiple calls to rand()

VII. Character arrays
    a. Have already used 1D arrays for text prompts
    b. Can also be 2D, but must be rectangular
    c. Simple types: take up constant amount of space per value, so allow immediate indexing
    d. Syntax comparison
    e. Example: remove all occurrences of a character