2D ARRAYS & FILES

CSSE 120—Rose Hulman Institute of Technology

Final Exam Facts

- □ Date: Thursday, November 19, 2009
- □ Time: 1:00 to 5:00 PM
- □ Venue: **G310 – G317 (see schedule page)**
- □ Chapters: Zelle chapters 1 to 9, 11:1-3, 11:6, Assigned C readings from Kochan plus Web resources linked from ANGEL Resources page
- □ On-computer part: will be only in C
- □ You may bring two double-sided sheets of paper this time.

Q1,2

Two-dimensional Arrays

- Like a list of lists in Python
- But size is fixed, like C arrays
- Visualize as a matrix:

NUM_COLS = 6 (loop using j)

NUM_ROWS = 3
(loop using i)

4	3	6	31	8	2
9	4	7	8	4	1
34	2	16	5	3	6

- Can make ragged arrays (different number of items in each row) but more difficult to do

2D Array Syntax

- □ Declaration reserves space, but doesn't set values to anything!

```
int nums[NUM_ROWS][NUM_COLS];
```

- □ Display

```
for (i = 0; i < NUM_ROWS; i++) {
    for (j = 0; j < NUM_COLS; j++) {
        printf("%2d ", nums[i][j]);
    }
    printf("\n");
}
```

Q3,4

Modify your code

- Ask the user for the number of rows and columns instead
- Then prompt them to input the value of each element
- Print out the values they entered in matrix form
- Challenge: print out the row sums and column sums

File handling

- Need to include <stdlib.h> to access many file handling functions
- Open a file using **fopen()**
- Modes:
  - “r” (read)
  - “w” (write)
  - “a” (append)
- Returns a file pointer to access the file: **FILE***
- Close a file using **fclose()**

Q5,6

A simple example

```
FILE *inFile;
inFile = fopen(“my_file.txt”, “r”);
if (inFile == NULL) {
   exit(EXIT_FAILURE);
}

// Read data from the file pointed to by inFile

fclose( inFile );
```

Q7,8

How do we read from a file?

- □ getc(my_fileptr) ; /* read the next character from the file*/
- □ fgets(buffer, n, my_fileptr); /* read the next line of text from file, up to **n-1** chars, into buffer */
- □ fscanf(my_fileptr, “%d”, &num); /* read the next int value from file into variable num*/

How do we write to a file?

- putc(c, my_fileptr) ; /* Converts int c to a char and write it to file */
- fputs(my_string, my_fileptr); /* Copies my_string to file, except the string terminating char */
- fprintf(my_fileptr, “%s\n”, my_string) ; /* Similar to printf() except the first parameter is a file pointer */

Q12

File Handling

- Check out ***FileDemo*** from your SVN repo
- See problem description in comments
- Work on solving problem for 10 minutes

Q13

HW27

- See instructions linked from ANGEL

- You'll read two 2D matrices from a file, perform math on them and then output the result back to a file.