2D ARRAYS & FILES

CSSE 120—Rose Hulman Institute of Technology

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];
```

- Looping through the array (to display its values)

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

Checkout **28-2DArrays**. Do its TODO's.

Q1,2

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()**

Q3,4

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);
```

Q5,6

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 */

Q10

File Handling

- Check out ***28-Files*** from your SVN repo
- Do its TODO's

- If you finish early, begin Homework 28