1. To be useful, we might store the scores for a whole class in a text file, where the first two values in the file are the number of students and number of scores for each student. Each student has a pair of lines consisting of a line with their name, and a line with their scores.

```
3 5
Doe, Jane
1 2 3 4 5
Morse, Cody
1.1 2.2 3.3 4.4 5.5
Smith, John
10 20 30 40 50
```

Two parallel, two-dimensional arrays can be used to store this information. *Parallel* arrays use the same subscript to store related information across multiple arrays. Conceptually, the information can be visualized as:

studentNames:

0	"Doe, Jane"
1	"Morse, Cody"
2	"Smith, John"
3	
99	

scores:

	0	1	2	3	4	5		49
0	1.0	2.0	3.0	4.0	5.0			
1	1.1	2.2	3.3	4.4	5.5			
2	10.0	20.0	30.0	40.0	50.0			
3								
99								

The declaration of these arrays are:

```
// Global Constant
const int CLASS_SIZE = 100;
const int MAX_SCORES = 50;
const int NAME_SIZE = 50;

int main() {
      double scores[CLASS_SIZE][MAX_SCORES];
      char fileName[NAME_SIZE],
           response[NAME_SIZE],
           studentNames[CLASS_SIZE][NAME_SIZE];
      int numberOfScores, numberOfStudents;
```

1. To be more useful, we might want to add another parallel array to label the columns (e.g., "HW 1", "HW 2", "Lab 1", "Lab 2", "Test 1").

a) Add this title information to the above file.

b) Draw the parallel array in the above picture of the arrays. Assuming this new array is call `columnTitles`.

c) Write the array definition for the array `columnTitles`.

d) The original (pre-titles) `readScoresFromFile` function is shown below. Make the modifications necessary to read and store the column titles.

```
void readScoresFromFile(char fileName[], char students[][NAME_SIZE],
                        double scores[][MAX_SCORES], int & studentCount, int & scoreCount) {
      int row, column;
      double score;
      ifstream inFile;

      inFile.open(fileName);
      if (!inFile) {
            cout << "Error opening file <" << fileName << "> for reading!";
      } else {
            inFile >> studentCount;
            inFile >> scoreCount;
            inFile.ignore(NAME_SIZE, '\n');

            for (row = 0; row < studentCount; row++) {
                  inFile.getline(students[row],NAME_SIZE);

                  for (column = 0; column < scoreCount; column++) {
                        inFile >> score;
                        scores[row][column] = score;

                  } // end for (column...

                  inFile.ignore(NAME_SIZE, '\n');
            } // end for (row...

            inFile.close();
      } // end if
} // end readScoresFromFile
```

e) Write the corresponding code for the `saveScoresToFile` function.