San José State University
Department of Computer Engineering

CMPE 180A
Algorithms and Data Structures in C++
Spring 2020
Instructor: Ron Mak

Assignment #4

Assigned: Tuesday, February 18
Due: Tuesday, February 25 at 5:30 PM
URL: http://codecheck.it/files/16091708133qzzdr5zkas8p3f0ib5zrzcwb
Canvas: Assignment 4. Student Scores
Points: 100

Student scores
This assignment will give you practice with dynamic arrays. You will first create a one-dimensional dynamic array of strings and a two-dimensional dynamic array of integers. Then you will access these arrays to print the data they contain.

Sample input file
Your program should read an input file **students.txt** that is similar in format to the following sample:
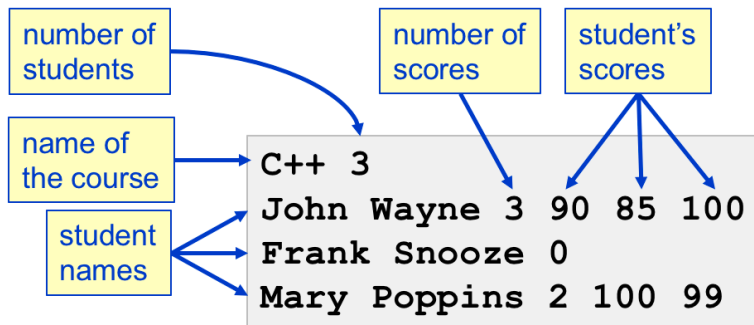
After the first line, the input file contains one line per student. Each student name is a first name and a last name only, and all the scores are integers. You may assume that there are no errors in the file; in particular, the number of students and the numbers of scores are all correct. There is at least one blank between the input elements.

Download the complete input file **students.txt**:
http://www.cs.sjsu.edu/~mak/CS144/assignments/4/students.txt
This file is already uploaded to CodeCheck.

Dynamic arrays
Create dynamic arrays to store the data from the input files:

- A one-dimensional dynamic string array to store the student names.

- A two-dimensional dynamic integer array (a matrix) to store the student scores. One row of the matrix stores the scores for one student, and there should be a row for each student that contains zero or more integer scores.

For this assignment, you must use dynamic arrays and not vectors. Use C++ strings for the names. Your arrays should conceptually be something like this:
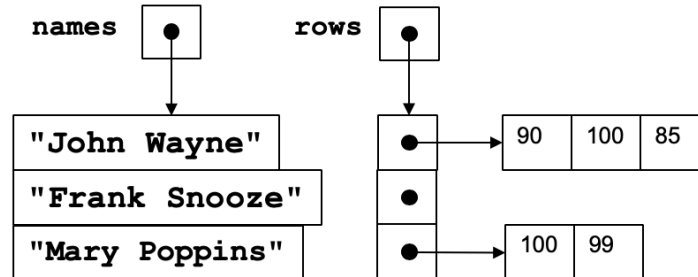
Remember that an array variable is actually a pointer to the first element of the array, and that a two-dimensional array is a one-dimensional array of pointers to the one-dimensional arrays of values. Therefore, the student scores matrix is dynamic both in the number of rows and in the number of scores per student. The matrix will be "ragged" since the rows will be different sizes.

You must delete your dynamic arrays before your program finishes executing.

Expected output
Here is the expected output format:

```
STUDENT SCORES for C++

John Wayne
    90    85   100

Frank Snooze
    (none)

Mary Poppins
   100    99
```

Your program must operate in this order:

1. Read all the input data and create the dynamic arrays.
2. Print the contents of the dynamic arrays.

Program structure
Your program structure should have good functional decomposition. You should have at least the following two functions:

- One function to read the input file and build the dynamic arrays.

- Another function to print the output from the content of the dynamic arrays, which are passed to it as arguments.

You may have other helper functions. Declare your functions with documentation before the main and define them after the main with internal documentation. Carefully decide which parameters should be passed by value or by reference, and which should be `const`.

Your program should have no global variables. Global constants are OK.

Rubrics

Criteria	Maximum points	
Good program output (as determined by CodeCheck)	**15**	
Dynamic arrays	**35**	
• Create and use dynamic arrays.		• 25
• Delete dynamic arrays.		• 10
Good program design	**45**	
• Function to build the dynamic arrays.		• 15
• Function to print the contents of the dynamic arrays.		• 15
• Good names and comments.		• 10
• No global variables.		• 5
Good program style	**5**	
• Function declarations and definitions.		• 5

You can submit as many times as necessary to get satisfactory results, and the number of submissions will not affect your score. When you're done with your program, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution.

Submit the signed zip file into Canvas: **Assignment 4. Student scores**

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.