

INTRODUCTION TO SCIENTIFIC AND ENGINEERING COMPUTATION

Assignment 3

June 2021

Objective: Deciding the outcome of an election.

Description

There is an election between four candidates A, B, C, and D. The voters are divided into a number of districts. The number of candidates is fixed but the number of districts is not known in advance. The votes are stored in a text file where each line contains the number of votes for each candidate in a district. For example, assume the following file contents:

```
192 48 206 37
147 90 312 21
186 12 121 38
114 21 408 39
267 13 382 29
```

This means that there are 5 districts and in district 1 (line 1) candidate A has 192 votes, candidate B has 48 votes, candidate C has 206 votes, and candidate D has 37 votes.

The candidate who gets more than 50% of the total votes wins the election. If no candidate has more than 50% of the votes, there will be a second round between the two candidates who have received the highest number of votes. The second round is not part of this assignment.

Assignment

Write a C program that will get the text file that contains the votes as a command line argument and prints the outcome.

Below is an example run with a winner in the first round:

```
./assignment3 votes.txt  
Winner: C
```

Another example where a second round will be needed:

```
./assignment3 votes.txt  
Second round: C and A
```

In the case of a second round, the letter of the candidate with the highest number of votes must come first. You don't need to validate the contents of the file; you can assume that there are a correct number of lines with a correct number of vote values on each line.

You have to use the following data structure to represent a candidate:

```
struct candidate {  
    char letter; // 'A', 'B', 'C', 'D'  
    int votes;  
    float percent;  
};
```

There is no limit on the number of districts; any array that requires this information must be dynamically allocated with the proper (minimal required) size after the number of districts is determined during the run of the program. The length of a line in the text file is at most 80; static allocation is allowed if you want to use an array for this purpose.

Write the following functions:

- A function that takes the input file name as a parameter and reads the vote data from the file. You can send other parameters to the function and return anything you want.
- A function that will calculate the total number of votes and percentages for each candidate.
- A function that will decide which candidate wins, or which two candidates will compete in the second round.

Rules

- Your source code file has to have the name "assignment3.c".
- Your program will be compiled using the following command on a Linux system. If it cannot be compiled and linked using this command, it will not be graded (failed submission).

```
gcc -std=c99 -Wall -Werror assignment3.c -o assignment3
```

- Your program will be checked using an automatic checker. Therefore, make sure you print the messages exactly as given in the example runs. You will be given a Calico test file for some basic I/O tests; run your assignment through Calico before submitting.
- Do NOT use statements for clearing the terminal or waiting for a keypress before exiting the program; these might cause your program to fail in the automatic tests. Some IDEs generate such statements, remove them. Running your program through Calico is the safest way to make sure that your program works as expected.
- Do NOT use any construct that hasn't been covered in the course before this week, such as arrays or functions. Do NOT use any C++ features such as `cout` and `cin`.
- Make sure your coding style is proper and consistent. Use the `clang-format` tool if necessary. Don't use any variable names in a language other than English.
- This is an individual assignment. Collaboration in any form is NOT allowed. No "working together", no sharing code in any form including showing code to your classmates to give them ideas.
- All the code you submit must be your own. Don't copy/paste any piece of code from any resource including anything you've found on the Internet.
- The assignments will be checked for plagiarism using both automated tools and manual inspection. Any assignment involving plagiarism and/or infringement of intellectual property will be not be graded and is subject to further disciplinary actions.