CPE 100 Introduction to Computer Programming
International Sections August 2020
Laboratory Exercise 5

Objective

This lab is intended to give you practice using parallel arrays and arrays of strings, as well as reviewing loops.

Instructions

Write a program called **classRank.c** which allows the user to enter a list of student names and student grades, then finds and prints the name and grade for the students with the highest and lowest grades. **Store the names in an array of strings (a two dimensional array of chars) as shown in Lecture 6. Store the grades in an array of integers.**

The maximum number of students is 20. The maximum length of a student name is 32 chars including the terminating zero. So the declaration of the names array might be something like this:

```
#define MAXSTUDENTS 20
#define MAXLEN 32

char studentNames[MAXSTUDENTS][MAXLEN];
```

Since every student will have a grade, the dimension of the array for the grades will be MAXSTUDENTS as well.

You will find two sample runs of the program below.

Run 1
How many students in the class (max is 20)? **5**

Name for student 1? **Harry**
Grade for student 1? **84**
Name for student 2? **Jenny**
Grade for student 2? **92**
Name for student 3? **Roger**
Grade for student 3? **43**
Name for student 4? **Lisa**
Grade for student 4? **78**
Name for student 5? **Robert**
Grade for student 5? **88**

Jenny had the highest grade in the class (92)
Roger had the lowest grade in the class (43)

Run 2
How many students in the class (max is 20)? **21**
Sorry, this program can handle no more than 20 students.

Pseudocode for computing a maximum value

```
set maxvalue to 0
for each item in the array
    if the value is greater than maxvalue
       set maxvalue = value
    endif
endfor
```

You can compute the maximum and minimum in the same loop. Be sure to initialize the minimum value to the largest possible value (100 in this case) before you start. **Also, check the max and min separately. Use two "if" statements, not "if" and "else".**

Extra challenge

If you would like to make the lab more challenging, add code to *validate* the name and the grade. For each name, check that the user has entered at least 2 characters. (Use the ***strlen()*** function to find out the number of characters.). If not, give an error message and ask for the name again. For each grade, check that the value entered is between 0 and 100 inclusive. If less than 0 or greater than 100, give an error message and ask again.

Upload your C source file only (classRank.c).