Course   Discussion   Syllabus   Earn a Professional Certificate   Weblinux   Wiki   Notes   Progress

Previous   Next

Activity: write numbers to a file

🔖 Bookmark this page

Taskgrader (External resource) (1.0 / 1.0 points)

| Problem statement | Hints | Submission | Solution |

You are grading final exams and you are recording grades (integers) in a file "myGrades.txt". You just got distracted by a phone call and can't remember whether or not you already recorded the grade of the student whose paper you just finished grading.

Expecting that this may not be the last time your phone rings you decide to write a C-program to help you out. The program should first read, from the user input, the grade (an integer) that you need to check on. Next, the program should open the file "myGrades.txt", find the last number in the file, close the file, and compare the last number from the file to the grade read from the user input. If the two grades are equal then you presumably already recorded the grade and don't have to do anything. If on the other hand the two grades do not equal each other then you have not yet recorded this new grade and need to do so by appending this new grade to the file.

Finally, reopen the file and print its new contents to the screen.

Things to consider:

(1) Be sure to separate grades in the file by printing one space between the grades.

(2) To test your program you need to create your own file titled "myGrades.txt" with grades stored as described. In weblinux you can do so by first typing "touch myGrades.txt" at the command prompt and then opening, editing and saving the file "myGrades.txt" in the file editor. When you submit your program to taskgrader we will provide an input file for testing purposes. It is thus crucially important that the file from which you are reading in your program is indeed named "myGrades.txt".

(3) You may assume that you have already recorded some grades in the file "myGrades.txt"; you therefore do not have to check whether the file indeed exists (although you may, of course, do so!)

Examples

User input:

```
83
```

Input file "myGrades.txt":

```
90 88 84 76
```

File "myGrades.txt" after program terminates:

Previous   Next