

CptS 122 – Data Structures

January 19, 2024



Your Name: _____
ID#: _____

TA's Name: _____
Section #: _____

SOLUTION: Take-Home: Quiz 2 (15 pts) - String Parsing

Using Canvas <https://canvas.wsu.edu/>, please submit your solution to the correct quiz folder. Your solution should be a .pdf file with the name <your last name>_quiz2.pdf and uploaded. To upload your solution, please navigate to your correct Canvas *lab* course space. Select the “Assignments” link in the main left menu bar. Navigate to the correct quiz submission folder. Click the “Start Assignment” button. Click the “Upload File” button. Choose the appropriate .pdf file with your solution. Finally, click the “Submit Assignment” button.

1. (15 pts) Write a C function called `fileLineParser()` that accepts a pointer to a `FILE` (name the parameter `inputFileStream`) and an array of `struct studentRecords` (name the parameter `arrStudentRecords`). The function reads the entire .csv file line-by-line, parses each line, and stores a copy of each value into the array of `struct studentRecords`. The function returns the *number* of records successfully parsed and stored in the array.

precondition: `inputFileStream != NULL`; all lines in the file are well formed in the required format.

The `struct studentRecord` is defined as follow:

```
typedef struct studentRecord
{
    char name[75]; // This will be used to store the first and last name
    int ID; // This will be used to store the student's ID number
    char major[15]; // This will be used to store the student's major
} StudentRecord;
```

Assume that each record in the .csv input file is formatted as follows:

```
Student name,ID number,major
```

PLEASE NOTE: you're not required to use `strncpy ()` in your solution!!!

```
int fileLineParser(FILE* inputStream, StudentRecord* arrStudentRecords)
{
    int recordCount = 0;
    char line[100] = "";

    while (fgets(line, 100, inputStream) != NULL) // are there more lines to read?
    {
        // yes, let's parse the line we read
        // separate the name and make sure that it doesn't exceed the allocated space for
        // the name field, which is 75 characters including the null character
        strncpy(arrStudentRecords[recordCount].name, strtok(line, ","), 75);
        // place a null character in the last element in the name in case
        // one was not read in the first 75 characters
        arrStudentRecords[recordCount].name[74] = '\0';
        // separate the ID and convert it to an integer
        arrStudentRecords[recordCount].ID = atoi(strtok(NULL, ","));
        // separate the major and make sure that it doesn't exceed the allocated space for
        // the major field, which is 15 characters including the null character
        strncpy(arrStudentRecords[recordCount].major, strtok(NULL, "\n"), 15);
        // place a null character in the last element in the name in case
        // one was not read in the first 15 characters
        arrStudentRecords[recordCount].major[14] = '\0';
        ++recordCount; // count the record
    }

    return recordCount;
}
```

Instructor: Andrew S. O'Fallon