

60-141 – Introduction to Programming II Winter, 2014

Assignment 4

Strings and Structures:

Write a C program that uses string processing functions to properly format a Comma Separated Values (csv) file of employee contacts and store them into an array of structures. The content of the input file ([input.csv](#)) is read by the program using input redirection technique that you already have used in the earlier assignments.

Assume that the employee structure has been defined as follows:

```
struct employee {
    int ID;
    char lastName[21];
    char firstName[21];
    char phone[20];
};

typedef struct employee Employee; //optional
```

Example:

Fields in the input file are comma separated as follows:

FirstName,LastName,AreaCode,PhoneNumber

Content of a sample input file:

```
monica,adam,822,5204614
harold,gartner,899,7147814
```

Corresponding field values in the array of structures (after formatting):

1	2
ADAM	GARTNER
Monica	Harold
(822) 520-4614	(899) 714-7814

Important:

- Once all data have been formatted and stored, your program should be able to:
 - Search for the information regarding an employee using either the employee ID or the last name as the search key.
 - Display all or any employee data.
 - Edit any employee data.
 - Add new employee.
- For formatting data, following points are to be considered:
 - Employee ID should be unique sequential integers starting from 1.
 - Last name must be all capital letters (upper case).
 - First character of the first name must be a capital letter. All other letters must be lower case.
 - Phone numbers must be store in the format as: (xxx) xxx-xxxx.

- e. The input file provided contains data for only 20 employees. Use an array of structures of size 40 for future addition of any new employee.

REQUIREMENTS:

- Write and document a complete C program that is capable of satisfying the requirements of this assignment problem.
- UNDOCUMENTED OR IMPROPERLY DOCUMENTED code will automatically lose 50% marks.
- PLAGIARIZED work will not be graded and receive a mark of ZERO and reported according to the Senate bylaws.
- The question requires use of I/O redirection. Please review the textbook for an example on using I/O redirection from flat files.
- TO SUBMIT: No later than the submission deadline, your email must be received by: cs14101@uwindsor.ca, late submissions are not accepted and will receive a mark of ZERO.
- Email your work as an attachment, include both the source file (assign1.c) and the script file (assign1.txt)
- see below how to create the script file.

To create a script file (one that logs your compilation steps and your output in a text file):

1. **script assign4.txt**
2. **cat assign2.c**
3. **cat input.txt**
4. **cc assign4.c**
5. **a.out < input.txt**
6. **ls -l**
7. **exit (DO NOT FORGET THIS STEP!!)**

Email both files with the mail Subject field format "**Ass #4 section 51**" (replacing 51 by your actual registered lab section number) to:

cs141@courses.cs.uwindsor.ca

NOTE: Submissions that are not received correctly by the deadline will automatically receive a ZERO mark. In the event that more than one email submission is sent, only the last one (according to the date and time stamp) will be marked.

It is your responsibility to ensure the email attachment is sent correctly (readable) and to the right mailbox by the deadline. If you omit the email subject header or fail to follow the format provided above for the attachment file, your assignment may not be graded.

Late assignment submissions are not accepted!

NOTES:

1. Your assignment must be RECEIVED by the due date and time. Late assignment submissions are NOT accepted. Keep your script file, and all your code unmodified as proof of its completion in case it is not received.
2. It is your responsibility to get an early start on the assignment, research and ask questions ahead of time from the due date.

3. You must use your uwindsor email account to submit your work. Please do not use other email accounts (hotmail, yahoo etc...) for the purpose of assignment submissions.
4. Marks will be deducted for unclear code. (improper spacing and alignment, hard to read programs and missing outputs).
5. Make sure you turn in a complete script file that clearly shows: your code, your compilation process, a listing of the directory showing your source file(s) and the a.out with the date/time stamps, and the output.
6. **PLAGIARISM: CHEATING IS NOT TOLERATED.** You must submit your own work. Students who are suspected of copying someone else's work will be reported to the department's chair and the Dean of Science and be dealt with in accordance with the University policies. You should not share your code with others. Codes that are similar to each other will **BOTH** be reported as potential evidence of copying. It is imperative that you write your own code.
7. Authorized/limited help on this assignment may be provided directly from your Lecture or Lab instructors and Teaching Assistants.