COSC 2436 hw1 Arrays and Files

1. **Introduction**
   You will create a C++ program to search through files and pull out the information needed. This should help the students with sorting through information from files on the servers and storing them in arrays.
2. **Input, Command, Output and example files**
   a. Input file
      - The input file will contain a list of books, that contains their title, the author and the year it was published. It can contain anywhere from 0 to 10000 books. Each book will be boxed in by brackets and there may be empty lines or unnecessary spacing.
      - There are four different categories (the categories must appear in this order for the information to be valid)
        a. genre
        b. title
        c. author
        d. year
      - In between each bracket there must be a title author and year in that order. If the categories are misspelled(case sensitive), missing or out of order then they aren't valid and shouldn't be added to your data. The category won't count as misspelled if it has a space in the middle of it

      - Examples of Correct data:

{genre:fantasy,title:Harry.Potter,author:J.K.Rowling,year:1997}
{genre:dy s top ian,t it le:Hunger .Ga me s,au t h o r:Suzanne.Collins, y e ar: 2008}
//Excess spaces are fine but should be removed in the output

      - Examples of Incorrect data:

{author:J.K.Rowling,title:Harry.Potter,year:1997} //out of order

      - Assumptions:
        - each category will be followed by a colon and will have some amount of letters and periods behind it
        - None of the words will have commas, colons or brackets in them. These symbols will only appear as they do in these examples, even in invalid data.
        - The only way data can be invalid is if the categories (genre, author, title, year) are missing or misspelled and they don't (this doesn't mean the actual title of the book is misspelled as that would be hard to implement)
        - So long as it fits this template the actual titles and and genres can be complete gibberish

- Duplicate information is allowed and should be repeated in the output file

b. Command file
    I. The command file will contain any number of the categories (I.e: title,author or year) followed by a colon and what it's looking for. You will then look through the books you have and output all the books that match all the categories in the command file.
    II. If the command file is empty then every single correct input should be printed
    III. If multiple of the same category are in the command file that means that any of as long as a book contains one of those names then it should be produced.
    IV. If every category is mentioned in the command file then the book must match with one of the specified categories in order to appear in the output
        A. EX: If the command file asks for three different years and three different titles then the book must match one of the titles **and** one of the years for it to show up in the output file.
c. Output file
    I. The output file will contain all the books specified by the command file in order they appeared in the input file (with the extra spaces removed).
    II. There should be no empty lines in this (the last line can be emptied and it won't change anything)
    III. The only books not mentioned should be the invalid one's from the input and books not specified in the command file
    IV. If there are no books that match the command file then the output file should be empty
d. Examples
    i. Ex1:

    input11.txt:

    {genre:fantasy,title:Harry.Potter,author:J.K.Rowling,year:1997}
    {genre:dystopian,t it le:Hunger .Ga me s,au t h o r:Suzanne.Collins, y e ar: 2008}
    {genre:fantasy,title:The.Lord.Of.The.Rings,author:J.R.R.Tolkien,year:1954}
    {genre:fantasy,author:Brandon.Sanderson,title:The.Way.Of.Kings,year:2010}

    command11.txt: (Empty)

    ans11.txt:

    {genre:fantasy,title:Harry.Potter,author:J.K.Rowling,year:1997}
    {genre:dystopian,title:Hunger.Games,author:Suzanne.Collins,year:2008}
    {genre:fantasy,title:The.Lord.Of.The.Rings,author:J.R.R.Tolkien,year:1954}

    Linux Command:

    ./files "input=input11.txt;command=command11.txt;output=output11.txt"

ii. Ex2:

input12.txt

{genre:fiction,title:Oliver.Twist,author:Cha rles.Dickens,year:18 38}
{genre:child.literature,title:the.cat.in.the. hat,author:Dr.Seus s,year:1957}
{genre:child.lit e rature,ti tle:Green.Eggs.And.H am,author:Dr.Seu ss,year:1960}

{genre:ficti on,title:Great.Expec tations,autho r:Char les.Dick ens,year:1861}
{genre:f ict i on,title:A.Christmas.Carol,author:Charles.D icke ns,year:1843}
{genre:fict ion ,title : The.House.Of.The.Dead,a uthor:Fyod or.Dostoevsky,year:1861}
{genre:fiction,titl e:The.Old.Man.And.The.Sea,author:Ernest.Hemingway,year:1957}

{g e n re: f ic tion , title:Gr eat.Exp ectation s,auth or:Charles.Dickens,year:1861}
{genre:chi ld.literatu r e, ti tl e:G reen.Eggs. A nd .Ha m ,auth or:Dr.Seuss,year:1960}

command12.txt

genre:fiction
year:1861
year:1957

ans12.txt

{genre:fiction,title:Great.Expectations,author:Charles.Dickens,year:1861}
{genre:fiction,title:The.House.Of.The.Dead,author:Fyodor.Dostoevsky,year:1861}
{genre:fiction,title:The.Old.Man.And.The.Sea,author:Ernest.Hemingway,year:1957}
{genre:fiction,title:Great.Expectations,author:Charles.Dickens,year:1861}

Linux Command:

./files “input=input12.txt;command=command12.txt;output=output12.txt”

iii. Ex3:

input13.txt

{genre:fiction,title:Oliver.Twist,author:Cha rles.Dickens,year:18 38}
{genre:child.literature,title:the.cat.in.the. hat,author:Dr.Seus s,year:1957}
{genre:child.lit e rature,ti tle:Green.Eggs.And.H am,author:Dr.Seu ss,year:1960}

{title:ficti on,genre:Great.Expec tations,autho r:Char les.Dick ens,year:1861}
{genrae:f ict i on,title:A.Christmas.Carol,author:Charles.D icke ns,year:1843}
{genred:fict ion ,title : The.House.Of.The.Dead,a uthor:Fyod or.Dostoevsky,year:1861}
{genremispelled:fiction,titl e:The.Old.Man.And.The.Sea,author:Ernest.Hemingway,year:1957}

command13.txt

title:Harry.Potter

ans13.txt (empty)

Linux Command:

./files “input=input13.txt;command=command13.txt;output=output13.txt”

3. **Requirements**

   Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc, will be treated as copy) will be detected and result in ”0” in this homework. The limit is 50% similarity. Here are some previous homework which have been found to copy each other (the main function has been deleted).

4. **Turn in your homework**

   Homework 1 needs to be turned in to our Linux server, follow the link here https://rizk.netlify.app/courses/cosc2430/2_resources/

   Make sure to create a folder under your root directory, name it hw1 (name need to be lower case), only copy your code to this folder, no testcase or other files needed.

   PS: This document may have typos, if you think something illogical, please email TAs for confirmation.