

Problem description:

The problem was reading in a file and writing out to a different file formatted output. Each line in the formatted file had to have the original and then the same words but with ' ; ' instead of ' , ' and each word capitalized. We also had to remove ".jpg" from the last word. We were tasked with doing this in three different languages: Python, Ruby, and C.

Solution:

My solutions for Python and ruby were very similar and probably a bit overcomplicated. The way I solved both is to split each line by the comma and fill that into a 2d array. So each array would be a sentence, and each index would be a word. I then had to check if the person had no middle name because if they did not, the file name would end up at the wrong index. If they didn't, I moved the filename to the next array index and set the previous one index it was in to nothing. For python, I filled a different array with each sentence replacing the comas, removing ".jpg" and capitalizing where needed, and then wrote each index of that array to a file. In ruby, I just wrote straight to the file line by line, replacing the comas, removing ".jpg" and capitalizing where needed. My solution for C differs heavily from the other two. In C I read the file line by line into a char[]. I then used a for loop to fill a different char[] character by character from the other char[]. I did this while replacing commas with semicolons, capitalizing the proper letters, and leaving out ".jpg". I then wrote out to the file both char arrays line by line.

## PYTHON:

```
PyLab7.py > ...
1  # Name: Joey Troyer
2  # Date: 10/28/22
3  # input: file "ConvertCSVinput.txt"
4  # output: write to a file to formatted output from the input file
5  # problem: Read in a txt with each word seperated by a coma and format it and write it to a different file
6  # algorithm: Read in file line by line split it by comas and storing it in a 2d array. Each array holds a
7  #             sentence and each index holds a word. If person has no middle then i have to swap row[1] and
8  #             row[2] and set row[1] to nothing. I write to the file each index in proper format, removing
9  #             the ".jpg" and capitalizing each word as needed.
10
11  import csv
12
13  with open('ConvertCSVinput.txt') as txt_file:
14      #reads in data splits by coma and fills into a 2d array
15      txt_reader = csv.reader(txt_file, delimiter=',')
16      out = []
17      jpg = ".jpg"
18      for row in txt_reader:
19
20          #check for no middle name
21          #if none move row[1] to row[2] and make row[1] empty
22          if any(jpg in word for word in row[:2]):
23              row.append(row[1])
24              row[1] = ""
25
26          #check if row[1] is empty or not
27          if(row[1]):
28              out.append(f"{row[0]},{row[1]},{row[2]} ; {row[0].capitalize()} ; {row[1].capitalize()} ; {row[2].removesuffix('.jpg').capitalize()} ;")
29          else:
30              out.append(f"{row[0]},{row[2]} ; {row[0].capitalize()} ; {row[2].removesuffix('.jpg').capitalize()} ;")
31
32  txt_file.close()
33
34  #write ti file formatted output
35  with open("Python_Output.txt",'w') as output:
36      for item in out:
37          output.write(f"{item}\n")
38
39  output.close()
```

## PYTHON OUTPUT:

```
Python_Output.txt
1  acosta,abel,f.jpg ; Acosta ; Abel ; F ;
2  albertson,dean,clinton.jpg ; Alberson ; Dean ; Clinton ;
3  albertson,dennis,frank.jpg ; Alberson ; Dennis ; Frank ;
4  allen,Judith,kathleen.jpg ; Allen ; Judith ; Kathleen ;
5  baird,josef,james.jpg ; Baird ; Josef ; James ;
6  baker,Tina,marie.jpg ; Baker ; Tina ; Marie ;
7  banegas,henry.jpg ; Banegas ; Henry ;
8  barclay,john,a.jpg ; Barclay ; John ; A ;
9  bean,lewis.jpg ; Bean ; Lewis ;
10 boykin,m.,arthur.jpg ; Boykin ; M. ; Arthur ;
11 cooper,shaun,hunter.jpg ; Cooper ; Shaun ; Hunter ;
12 cyrs,thomas,earl.jpg ; Cyrs ; Thomas ; Earl ;
13 dage,erik,l.jpg ; Dage ; Erik ; L ;
14 davis,Janet,lynn.jpg ; Davis ; Janet ; Lynn ;
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

```
PS C:\Users\super\OneDrive\Desktop\CS471\Lab7> python3 .\PyLab7.py
PS C:\Users\super\OneDrive\Desktop\CS471\Lab7> 
```

## RUBY:

```
Lab7 > RubyLab7.rb
2 # Date: 10/20/22
3 # input: file "ConvertCSVinput.txt"
4 # output: write to a file to formatted output from the input file
5 # problem: Read in a txt with each word separated by a comma and format it and write it to a different file
6 # algorithm: Read in file line by line split it by commas and storing it in a 2d array. Each array holds a
7 #             sentence and each index holds a word. If person has no middle then i have to swap row[1] and
8 #             row[2] and set row[1] to nothing. I write to the file each index in proper format, removing
9 #             the ".jpg" and capitalizing each word as needed.
10
11 input = "ConvertCSVinput.txt"
12 output = "Ruby_Output.txt"
13
14 #reads in data splits by comma and fills into a 2d array
15 data = File.readlines(input)[0..-1].map { |line|
16   line.chomp.split(",").map(&:to_s)
17 }
18
19 #checks for no middle name
20 #if none move row[1] to row[2] and make row[1] empty
21 for item in data
22   if(!item[2])
23     item[2] = item[1]
24     item[1] = ''
25   end
26 end
27
28 # write to file
29 File.open(output, "w") { |f|
30   for item in data
31     #checks if theres a middle name
32     if(item[1] != '')
33       f.write("#{item[0]},#{item[1]},#{item[2]} ; #{item[0].capitalize()} ; #{item[1].capitalize()} ; #{item[2].chomp(".jpg").capitalize()} ;\n")
34     else
35       f.write("#{item[0]},#{item[2]} ; #{item[0].capitalize()} ; #{item[2].chomp(".jpg").capitalize()} ;\n")
36     end
37   end
38 }
```

## RUBY OUTPUT:

```
Lab7 > Ruby_Output.txt
1  acosta,abel,f.jpg ; Acosta ; Abel ; F ;
2  albertson,dean,clinton.jpg ; Albertson ; Dean ; Clinton ;
3  albertson,dennis,frank.jpg ; Albertson ; Dennis ; Frank ;
4  allen,judith,kathleen.jpg ; Allen ; Judith ; Kathleen ;
5  baird,josef,james.jpg ; Baird ; Josef ; James ;
6  baker,tina,marie.jpg ; Baker ; Tina ; Marie ;
7  banegas,henry.jpg ; Banegas ; Henry ;
8  barclay,john,a.jpg ; Barclay ; John ; A ;
9  bean,lewis.jpg ; Bean ; Lewis ;
10 boykin,m.,arthur.jpg ; Boykin ; M. ; Arthur ;
11 cooper,shaun,hunter.jpg ; Cooper ; Shaun ; Hunter ;
12 cyrs,thomas,earl.jpg ; Cyrs ; Thomas ; Earl ;
13 dage,erik,l.jpg ; Dage ; Erik ; L ;
14 davis,janet,lynn.jpg ; Davis ; Janet ; Lynn ;
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   JUPYTER

```
jtroyer@granville:~/CS471/Lab7> ruby RubyLab7.rb
jtroyer@granville:~/CS471/Lab7> 
```

C:

```
Lab7 > C Lab7.c > main()
1 // Name: Joey Troyer
2 // Date: 10/31/22
3 // input: file "ConvertCSVinput.txt"
4 // output: write formatted output to a "C_Output.txt" from the "ConvertCSVinput.txt"
5 // problem: Read in a txt with each word seperated by a coma and format it and write it to a different file
6 // algorithm: Read in file line by line into a char[]. Use a for loop to fill a different char[] character by
7 //             character while replacing ; and capitalizing the proper letters. Write out both char[] to file
8 //             line by line.
9
10
11 #include <stdio.h>
12 #include <string.h>
13 #include <ctype.h>
14
15 int main() {
16     char line[256];
17     FILE* readFile = fopen("ConvertCSVinput.txt", "r");
18     FILE* writeFile = fopen("C_Output.txt", "w");
19     int flag = 0;
20
21     //read in file line by line and stores in var "line"
22     while (fgets(line, sizeof(line), readFile)) {
23
24         //different iterator variable for edit array
25         //to handle different string sizes
26         int x = 0;
27
28         //char array to fill with formatted data
29         //initializes a new char array every loop to get rid of previous data
30         char edit[256] = "";
31
32         //removes all new lines from var "line"
33         line[strcspn(line, "\n")] = 0;
34
35         //capitalizes first char in edit
36         edit[x++] = toupper(line[0]);
37
38         //for loop that fills edit char by char
39         //iterates through the length of line minus 5 so we dont copy over ".jpg"
40         for(int i = 1; i <= (strlen(line) - 5); i++) {
41
42             //if theres a coma add " ", ";", " " to each next array index of edit
43             //set flag = 1 because we know the next char needs to be capitalized
44             if(line[i] == ',') {
45                 edit[x++] = ' ';
46                 edit[x++] = ';';
47                 edit[x++] = ' ';
48                 flag = 1;
49             }
50
51             //Set incoming char to uppercase
52             else if(flag == 1) {
53                 edit[x++] = toupper(line[i]);
54                 flag = 0;
55             }
56             else {
57                 //just copy data from line into next index of edit
58                 edit[x++] = line[i];
59             }
60         } //end for
61
62         //write to file the formatted data line by line
63         fprintf(writeFile, "%s ; %s \n", line, edit);
64     } //end while
65     return 0;
66 } //end main
```

## C OUTPUT:

```
Lab7 > ≡ C_Output.txt
 1  acosta,abel,f.jpg ; Acosta ; Abel ; F ;
 2  alberson,dean,clinton.jpg ; Alberson ; Dean ; Clinton ;
 3  alberson,dennis,frank.jpg ; Alberson ; Dennis ; Frank ;
 4  allen,Judith,kathleen.jpg ; Allen ; Judith ; Kathleen ;
 5  baird,josef,james.jpg ; Baird ; Josef ; James ;
 6  baker,Tina,marie.jpg ; Baker ; Tina ; Marie ;
 7  banegas,henry.jpg ; Banegas ; Henry ;
 8  barclay,john,a.jpg ; Barclay ; John ; A ;
 9  bean,lewis.jpg ; Bean ; Lewis ;
10  boykin,m.,arthur.jpg ; Boykin ; M. ; Arthur ;
11  cooper,shaun,hunter.jpg ; Cooper ; Shaun ; Hunter ;
12  cyrs,thomas,earl.jpg ; Cyrs ; Thomas ; Earl ;
13  dage,erik,l.jpg ; Dage ; Erik ; L ;
14  davis,Janet,lynn.jpg ; Davis ; Janet ; Lynn ;
```

---

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER
```

```
jttroyer@granville:~/CS471/Lab7> gcc Lab7.c -o Lab7
jttroyer@granville:~/CS471/Lab7> ./Lab7
jttroyer@granville:~/CS471/Lab7> █
```