

Name: Pratyay Kumar
Aggie id: 800811773
Date: 10/31/2022
Lab: 7

Program #7: Convert input file into formatted Semi Colon Separated file

Problem Description: We must write three programs that do the same task. Which takes a file name that contains first name, last name, and sometimes middle name. And convert the input into a formatted semicolon-separated file.

Language	Algorithm
C	Step1: stored the character 'ch' from the input file to current_line. Step2: for each iteration, current_line appends and finally stores the current line of the input in the form of a string. Step3: Store the character 'ch' in the current_line string If ch == ',' then the current_line is modified as per the requirements. Step4: finally, the required character from the output is capitalized.
Python	Step1: current_name stores the first, middle, and last name from the input file Step2: capitalize the characters as per the requirement Step3: split the last name with jpg
Ruby	The algorithm in ruby is same as in Python Step1: current_name stores the first, middle, and last name from the input file Step2: capitalize the characters as per the requirement Step3: split the last name with jpg

Language	Code
C	<pre>/* * Author: Pratyay Kumar * Date: 10/31/2022 * Program Description: Converting input file into formatted Semi Colon Separated file * Language: C */ #include <stdio.h> #include <string.h> int main() { // Taking in the input file FILE *input = fopen("ConvertCSV input","rb"); // Writing the output file in cOutput.txt FILE *output = fopen("cOutput.txt","w"); if (input == NULL) { printf ("Error"); } // Initializing the variables char ch = fgetc(input); char part1[20] = {}, part2[20] = {}, part3[20] = {}; int index = 0; // current_line will store in each line from the input file char current_line[100] = {}; // Algorithm: (Rough idea of the algorithm)</pre>

```

// Step1 : stored the character 'ch' from the input file to current_line.
// Step2 : for each iteration, current_line appends, and finally stores
//          current line of the input in form of string.
// Step3 : while storing the character 'ch' in current_line string, if ch == ',',
//          then current_line is modified as per the requirements.
// Step4 : finally, the required characters from the output is capitalized.
while ( ch != EOF ) {
    // storing " ; " whenever ',' comes
    if ( ch == ',' ) {
        current_line[index] = ' ';
        current_line[index+1] = ';';
        current_line[index+2] = ' ';
        index = index + 3;
    }
    else { // storing the characters from the input file
        current_line[index] = ch;
        index = index + 1;
    }

    // printing as per the requirements (FILENAME ';')
    if ( ch != '\n' ) {
        printf ("%c", ch);
        fprintf(output, "%c", ch); // write to file
    }
    else {
        printf (" ; ");
        fprintf(output, " ; ");
    }

    // length of the current line
    int l = strlen(current_line);

    if ( ch == '\n' ){
        // capatilize the first character of the string

```

```

        current_line[0] = current_line[0]-32;

        // this loop removes .jpg from the end of every string, and capitalize the characters as per
the requirements
        for ( int i=0; i<l-3; i++ ) {
            // capitalizing
            if ( current_line[i] == ';' && (int)current_line[i+2]>= 97)
                current_line[i+2] = current_line[i+2] - 32;
            // removing .jpg
            if ( current_line[i] == '.'&& current_line[i+1]=='j'&& current_line[i+1]=='j'&&
current_line[i+2]=='p'&& current_line[i+3]=='g') {
                current_line[i] = '\0';
                current_line[i+1] = '\0';
                current_line[i+2] = '\0';
                current_line[i+3] = '\0';
            }
        }

        // printing the modified string
printf("%s ;\n", current_line);

        // writing the string in an output file
fprintf(output, "%s ;\n", current_line);

        // re-initializing current_line and index for next line
memset(current_line, 0, sizeof current_line);
index = 0;
    }

    // appending the character to next character
ch = fgetc(input);

}

```

	<pre>// closing input and output files fclose(input); fclose(output); }</pre>
Python	<pre># /* # * Author: Pratyay Kumar # * Date: 10/31/2022 # * Program Description: Converting input file into formatted Semi Colon Separated file # * Language: Python # */ # Taking in the input file input = open("ConvertCSV input", 'r') # Writing in output file output = open('pythonOutput.txt', 'w') # reading line by line the input file line = input.readlines() # pointer to the line p = 0 # Algorithm: # Step1: current_name stores the first, middle and last name from the input file # Step2: capitalize the characters as per the requirement # Step3: split last name with jpg for i in line: # current_name store the list of first, middle and last name current_name = line[int(p)].split(",") # counting if the input contains all the part of the name, or 2 part or just one part of name. counter = line[int(p)].count(',') # if the input is given for all three part of names if counter == 2: # capitalizing the first character of names temp1 = current_name[0].capitalize() temp2 = current_name[1].capitalize()</pre>

```

# removing \n character from the line
temp3 = current_name[2].strip()
# splitting jpg
temp3 = temp3.split(".")
temp3 = temp3[0].capitalize()
# printing on screen
print(current_name[0]+","+current_name[1]+","+current_name[2].strip()+" ; ", end = "")
print(temp1 + " ; " + temp2 + " ; " + temp3 + " ;", end="\n")
# writing the printed stuff in the file
output.write(current_name[0]+","+current_name[1]+","+current_name[2].strip()+" ; ")
output.write(temp1 + " ; " + temp2 + " ; " + temp3 + " ;" + "\n")
# if the input is given for 2 parts of the name ending with .jpg
if counter == 1:
    # capitalizing the first character of names
    temp1 = current_name[0].capitalize()
    temp2 = current_name[1].split(".")
    temp2 = temp2[0].capitalize()
    # printing and removing \n character
    print(current_name[0]+","+current_name[1].strip()+" ; ", end = "")
    print(temp1 + " ; " + temp2 + " ;", end="\n")
    # writing in the file
    output.write(current_name[0]+","+current_name[1].strip()+" ; ")
    output.write(temp1 + " ; " + temp2 + " ;" + "\n")
# if the input is given for only 1 part of the name ending with .jpg
if counter == 0:
    # capitalizing the first character of names
    current_line = line[int(p)].split(".")
    temp = current_line[0].capitalize()
    # printing and removing \n character
    print(line[p].strip() + " ; ", end="")
    print(temp + " ;", end="\n")
    # writing in the file
    output.write(line[p].strip() + " ; ")
    output.write(temp + " ; " + "\n")

```

	<pre> # appending the line to next line p = p + 1 </pre>
Ruby	<pre> # /* # * Author: Pratyay Kumar # * Date: 10/31/2022 # * Program Description: Converting input file into formatted Semi Colon Separated file # * Language: Ruby # */ # Taking in the input file input = File.open("ConvertCSV input") # Writing in output file output = File.open("rubyOutput.txt", "w") # reading line by line the input file line = input.readlines.map(&:chomp) l = line.length() # pointer to the line p = 0 # Algorithm: # Step1: current_name stores the first, middle and last name from the input file # Step2: capitalize the characters as per the requirement # Step3: split last name with jpg for i in 0..l do # checking if the current line is not nil if line[p] != nil # counting if the input contains all the part of the name, or 2 part or just one part of name. counter = line[p].count(",") # if the input is given for all three part of names if counter == 2 # current_name store the list of first, middle and last name current_line = line[p].split(",") # splitting with jpg temp = current_line[2].split(".") end end end </pre>

```

        # capitalizing the first character of names
        temp = temp[0].capitalize()
        # printing on screen
        print (current_line[0] + "," + current_line[1] + "," + current_line[2] + " ; ")
        print (current_line[0].capitalize() + " ; " + current_line[1].capitalize() + " ; " + temp + "
;")

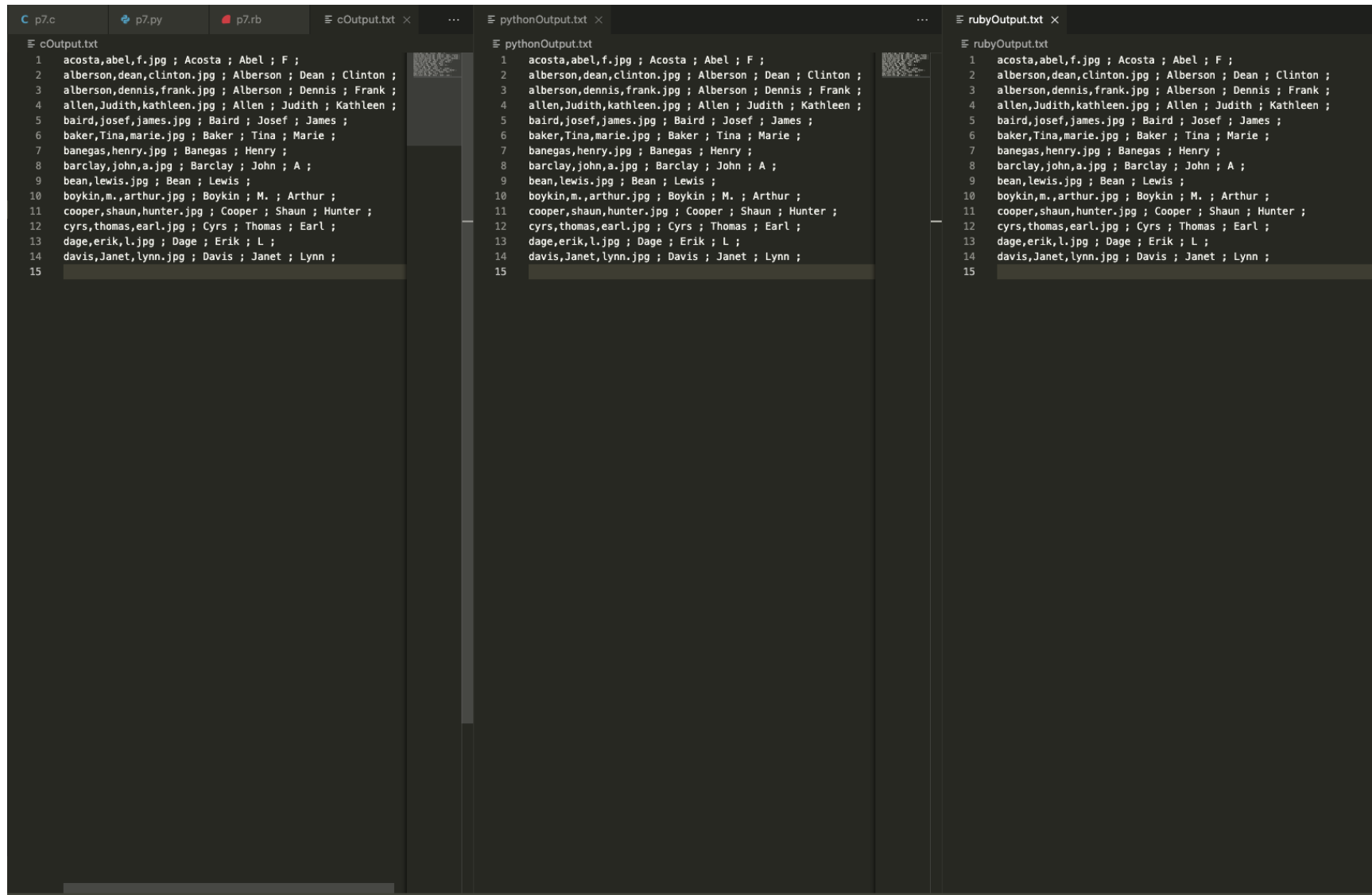
        puts("\n")
        # storing in file
        output.write(current_line[0] + "," + current_line[1] + "," + current_line[2] + " ; ")
        output.write(current_line[0].capitalize() + " ; " + current_line[1].capitalize() + " ; " +
temp + " ;")
        output.write("\n")
    end
    # if the input is given for 2 parts of the name ending with .jpg
    if counter == 1
        # current_name store the list of name available
        current_line = line[p].split(",")
        # splitting with jpg
        temp = current_line[1].split(".")
        # capitalizing the first character of names
        temp = temp[0].capitalize()
        # printing
        print (current_line[0] + "," + current_line[1] + " ; ")
        print (current_line[0].capitalize() + " ; " + temp + " ;")
        puts("\n")
        #writing in file
        output.write(current_line[0] + "," + current_line[1] + " ; ")
        output.write(current_line[0].capitalize() + " ; " + temp + " ;")
        output.write("\n")
    end
    # if the input is given for only 1 part of the name ending with .jpg
    if counter == 0
        # removing jpg
        current_line = line[p].split(".")

```



```
        # capitalizing
        temp = current_line[0].capitalize()
        # printing
        print (line[p] + " ; ")
        print (temp + " ;")
        puts ("\n")
        # writing in file
        output.write(line[p] + " ; ")
        output.write(temp + " ;")
        output.write("\n")
    end
end
# appending the index of current line
p = p + 1
end
```

The output of the above codes: cOutput, pythonOutput, rubyOutput



```
cOutput.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 ;
15

pythonOutput.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 ;
15

rubyOutput.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 ;
15
```

Link to the code can be found here: https://drive.google.com/drive/folders/1vlxvyvPs378QZ-96vQ4zh0_bhwAxo81q?usp=sharing