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

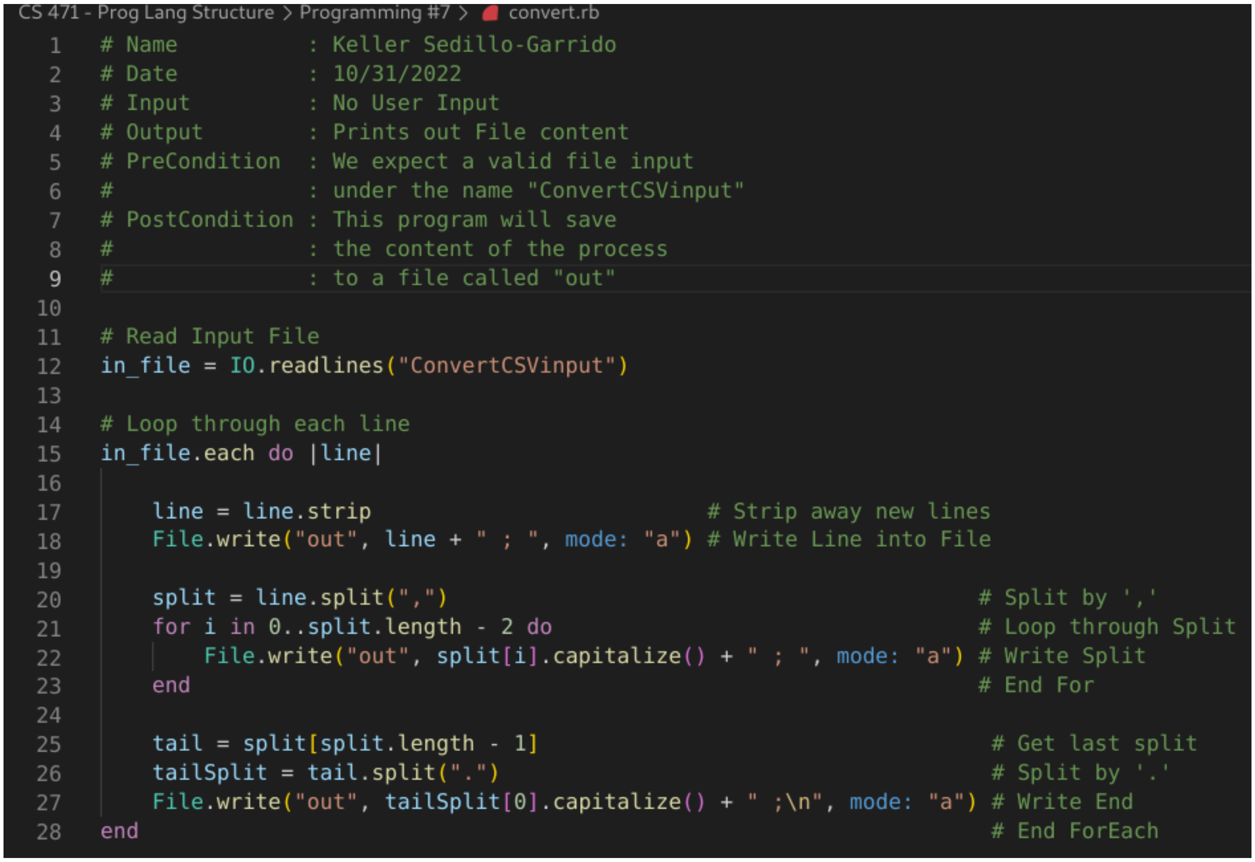
Keller Sedillo-Garrido

Problem Description:

We have been given a file which contains names of filles in it. We are to program in three different languages that are able to parse the files and output the final result into an output file. The format in the output file will look like “FILENAME ';' Last Name ';' FirstName ';' Middle Name”

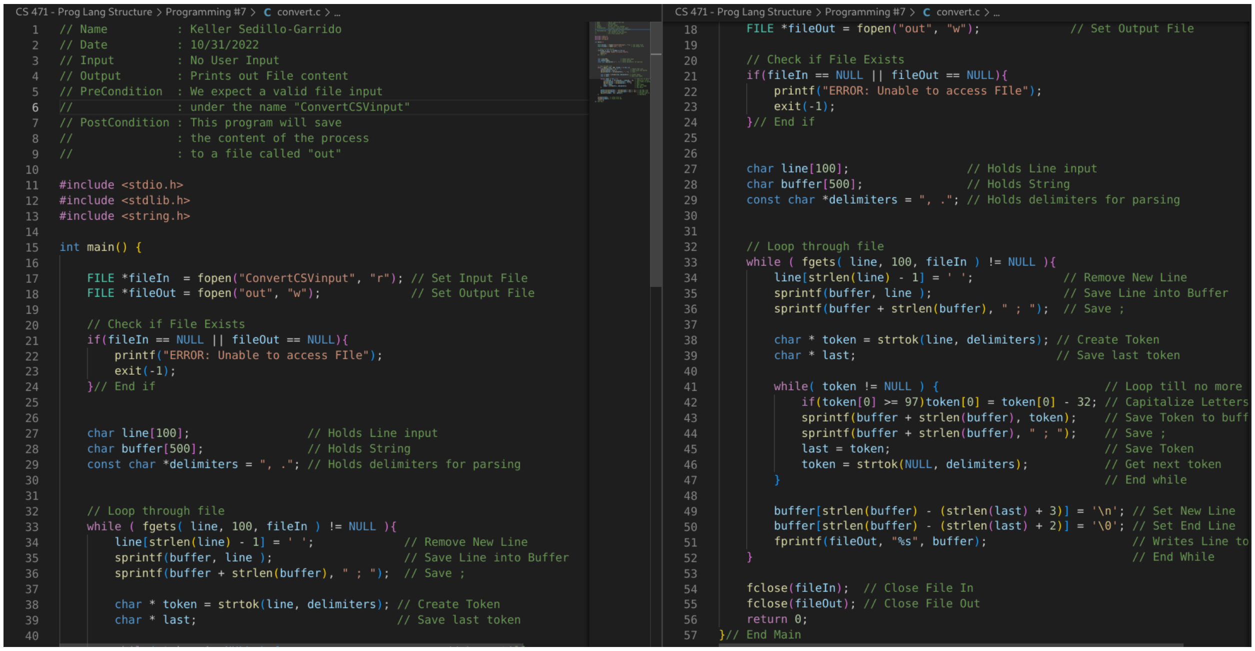
Programs:

Ruby:



```ruby
# Name           : Keller Sedillo-Garrido
# Date           : 10/31/2022
# Input          : No User Input
# Output         : Prints out File content
# PreCondition   : We expect a valid file input
#                : under the name "ConvertCSVinput"
# PostCondition  : This program will save
#                : the content of the process
#                : to a file called "out"

# Read Input File
in_file = IO.readlines("ConvertCSVinput")

# Loop through each line
in_file.each do |line|

    line = line.strip                          # Strip away new lines
    File.write("out", line + " ; ", mode: "a") # Write Line into File

    split = line.split(",")                                         # Split by ','
    for i in 0..split.length - 2 do                                 # Loop through Split
        File.write("out", split[i].capitalize() + " ; ", mode: "a") # Write Split
    end                                                             # End For

    tail = split[split.length - 1]                                   # Get last split
    tailSplit = tail.split(".")                                      # Split by '.'
    File.write("out", tailSplit[0].capitalize() + " ;\n", mode: "a") # Write End
end                                                                  # End ForEach
```

Java:



CS 471 - Prog Lang Structure > Programming #7 > C convert.c > ...

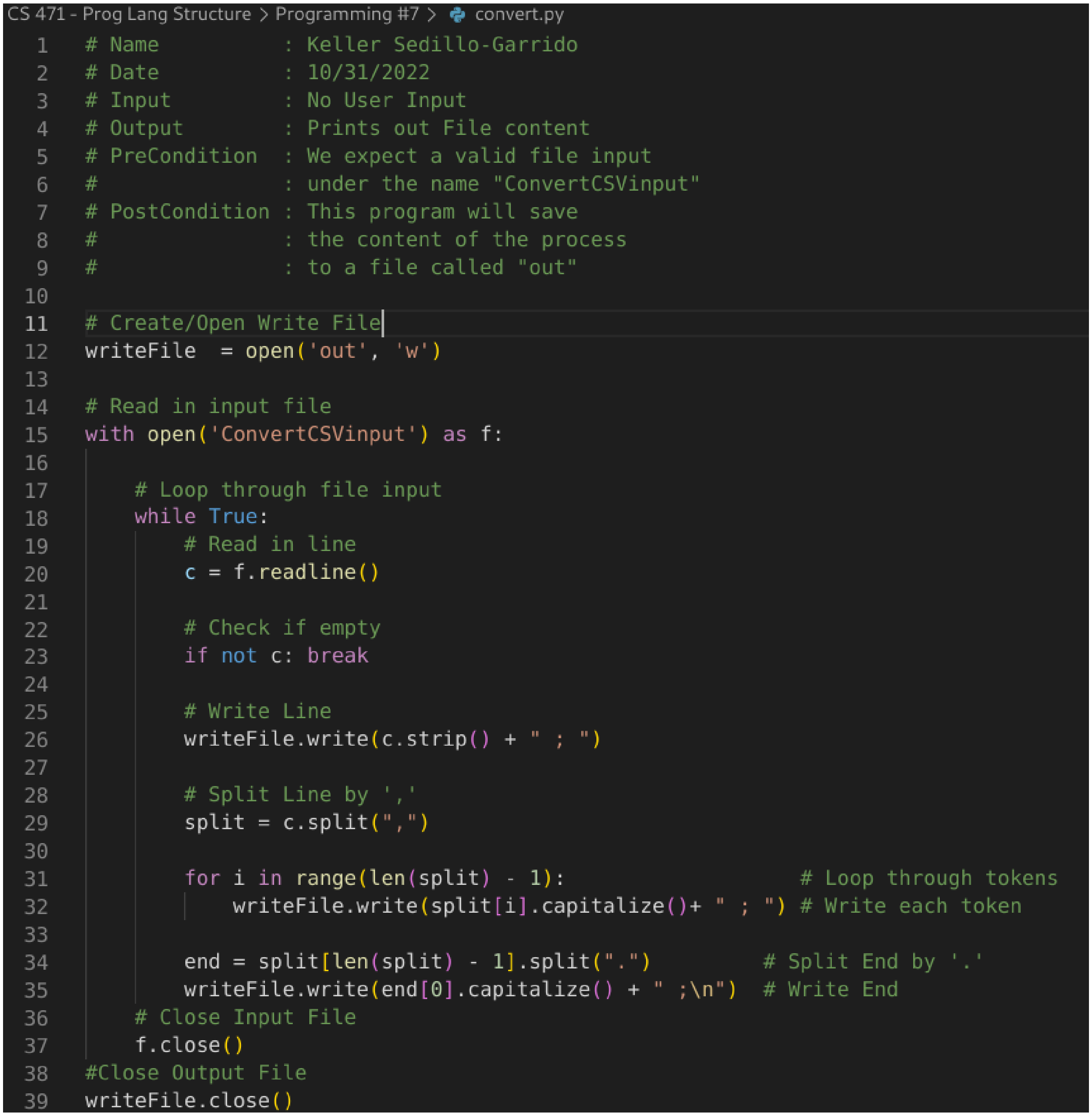
```c
// Name           : Keller Sedillo-Garrido
// Date           : 10/31/2022
// Input          : No User Input
// Output         : Prints out File content
// PreCondition   : We expect a valid file input
//                : under the name "ConvertCSVinput"
// PostCondition  : This program will save
//                : the content of the process
//                : to a file called "out"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {

    FILE *fileIn  = fopen("ConvertCSVinput", "r"); // Set Input File
    FILE *fileOut = fopen("out", "w");             // Set Output File

    // Check if File Exists
    if(fileIn == NULL || fileOut == NULL){
        printf("ERROR: Unable to access FIle");
        exit(-1);
    }// End if


    char line[100];                 // Holds Line input
    char buffer[500];               // Holds String
    const char *delimiters = ", ."; // Holds delimiters for parsing


    // Loop through file
    while ( fgets( line, 100, fileIn ) != NULL ){
        line[strlen(line) - 1] = ' ';             // Remove New Line
        sprintf(buffer, line );                   // Save Line into Buffer
        sprintf(buffer + strlen(buffer), " ; ");  // Save ;

        char * token = strtok(line, delimiters); // Create Token
        char * last;                             // Save last token

```

CS 471 - Prog Lang Structure > Programming #7 > C convert.c > ...

```c
    FILE *fileOut = fopen("out", "w");             // Set Output File

    // Check if File Exists
    if(fileIn == NULL || fileOut == NULL){
        printf("ERROR: Unable to access FIle");
        exit(-1);
    }// End if


    char line[100];                 // Holds Line input
    char buffer[500];               // Holds String
    const char *delimiters = ", ."; // Holds delimiters for parsing


    // Loop through file
    while ( fgets( line, 100, fileIn ) != NULL ){
        line[strlen(line) - 1] = ' ';             // Remove New Line
        sprintf(buffer, line );                   // Save Line into Buffer
        sprintf(buffer + strlen(buffer), " ; ");  // Save ;

        char * token = strtok(line, delimiters); // Create Token
        char * last;                             // Save last token

```

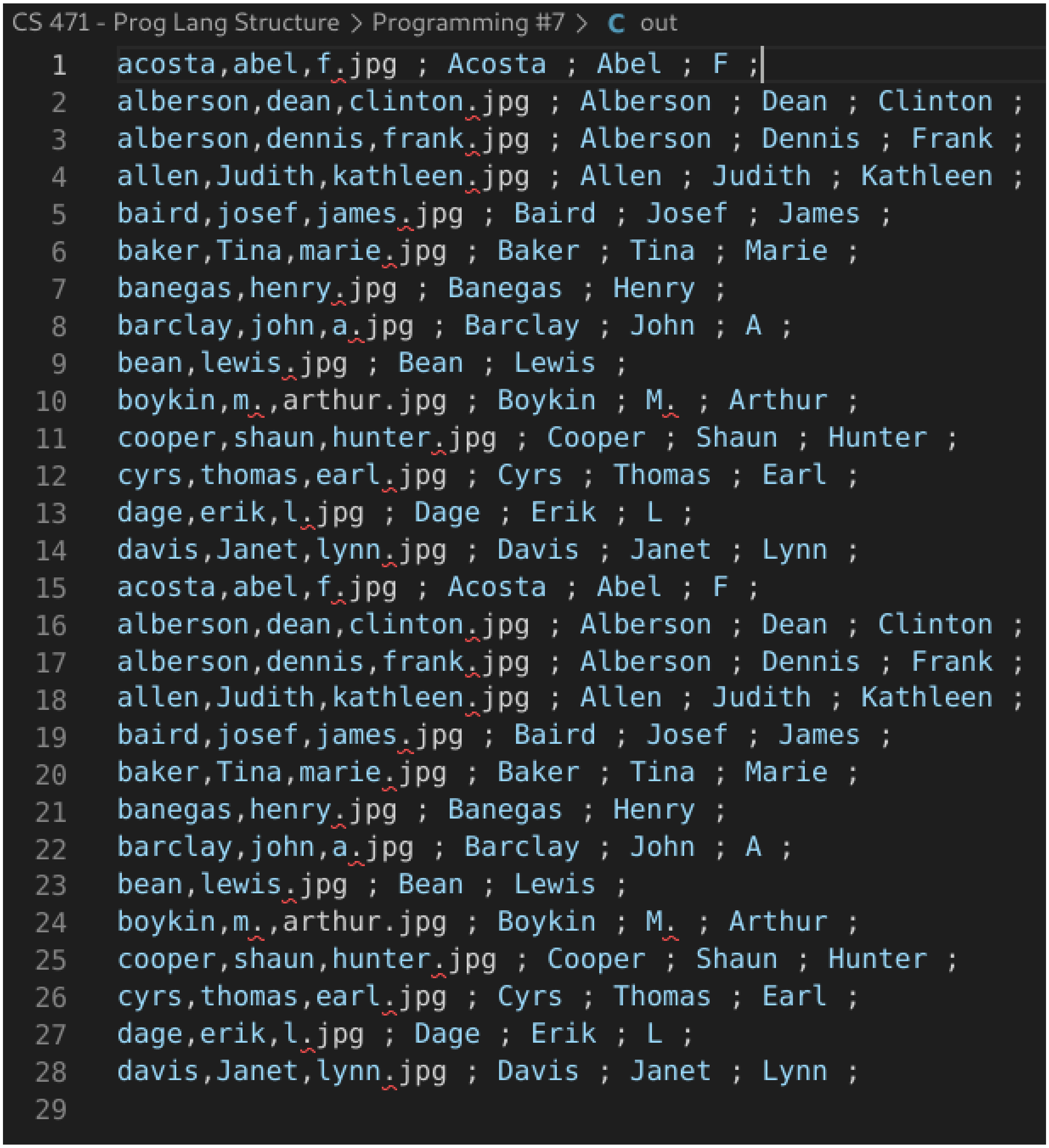
```c
        while( token != NULL ) {                           // Loop till no more
            if(token[0] >= 97)token[0] = token[0] - 32; // Capitalize Letters
            sprintf(buffer + strlen(buffer), token);    // Save Token to buff
            sprintf(buffer + strlen(buffer), " ; ");    // Save ;
            last = token;                               // Save Token
            token = strtok(NULL, delimiters);           // Get next token
        }                                               // End while

        buffer[strlen(buffer) - (strlen(last) + 3)] = '\n'; // Set New Line
        buffer[strlen(buffer) - (strlen(last) + 2)] = '\0'; // Set End Line
        fprintf(fileOut, "%s", buffer);                     // Writes Line to
    }                                                       // End While

    fclose(fileIn);  // Close File In
    fclose(fileOut); // Close File Out
    return 0;
}// End Main
```

Python:



```python
# Name          : Keller Sedillo-Garrido
# Date          : 10/31/2022
# Input         : No User Input
# Output        : Prints out File content
# PreCondition  : We expect a valid file input
#               : under the name "ConvertCSVinput"
# PostCondition : This program will save
#               : the content of the process
#               : to a file called "out"

# Create/Open Write File
writeFile  = open('out', 'w')

# Read in input file
with open('ConvertCSVinput') as f:

    # Loop through file input
    while True:
        # Read in line
        c = f.readline()

        # Check if empty
        if not c: break

        # Write Line
        writeFile.write(c.strip() + " ; ")

        # Split Line by ','
        split = c.split(",")

        for i in range(len(split) - 1):                   # Loop through tokens
            writeFile.write(split[i].capitalize()+ " ; ") # Write each token

        end = split[len(split) - 1].split(".")         # Split End by '.'
        writeFile.write(end[0].capitalize() + " ;\n")  # Write End
    # Close Input File
    f.close()
#Close Output File
writeFile.close()
```

Output:



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

```

Conclusion:

We see that all three languages are able to accomplish the task at hand. However, we see that it's much easier to program in Ruby and Python than C. Personally, I think this is the case because C has a more rigid string system at the cost of poor writability.