Electronic System School - ELSYS

EK (Error Killers)

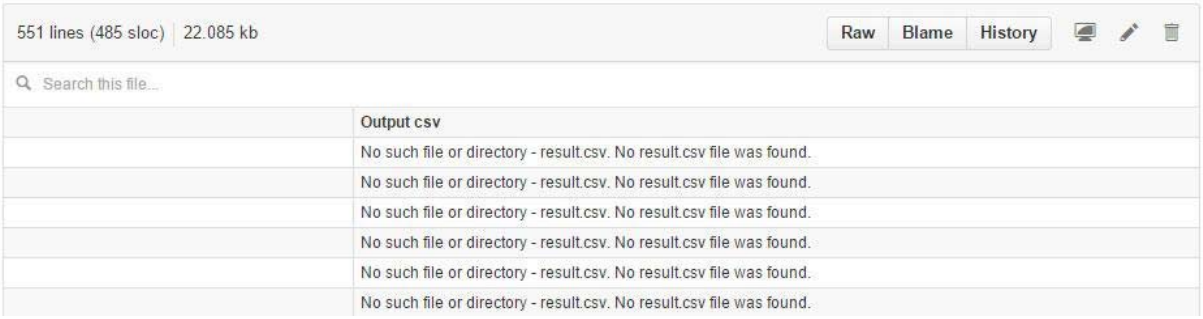
Elena Karakoleva

and

Kamena Dacheva

October, 2014

There were a lot of different errors. One of them was the wrong named SCV file. The file should be named “result.csv” but a big part of the students named it “result**s.**csv”This mean that the students don’t read what they have to do very carefully. The only way this problem to be solved is everybody to read more than ones their tasks and to make sure they check all the points in it.



Beside that error there are a lot of others different errors that the students from ELSYS had made.

For example very common error in the programs was the syntax error. A lot of students made mistakes in splitting strings.

Other common error was the sorting. There were a lot of students that sort by key when according to the task they must sort by value or the opposite. Also there were students that don’t sort their results at all.

According to our statistic the average errors are around 5 in program. There were all sort of problems. Starting with wrong name of the CSV file and ending with errors that don’t have quick solution. From 34 program only 8 are with correct results.

The errors can be categories mainly in syntax errors, errors that were made because the students didn’t read their tasks very carefully and errors that were made because some of the students hurried or get worried and they probably couldn’t think straight.

All of those errors can be avoided.

First of all our team want to suggest all of the students from ELSYS and from another school to read very carefully what they have to do. It is really important to fully understand the task that is given to you. Read it more than once. Once is never enough. Start writing your code when you are sure that you know what your task want from you.

Also before starting the program take a deep breath and don’t panic.

You should learn to manage your time properly otherwise the time will never be enough for you.

When you finish your program read the task again and check your program.

Appendexes

11a class

Ivelin Slavchev

```
1. =begin
2.    Develop a program named FirstName_LastName_ClassNumber_835552.rb
3.
4. 1. you are given two arguments for a folders with files;
5. 1.1 if there are other arguments they should be discarded;
6. 2. Find all the files from both folders that are not in the format
   FirsrName_LastName_digits.rb. If there are duplicates the file must be written only once. If
   two files are of the same lenght those files should be sorted in ASC order;
7. 3. Calculate the length of their names (including extensions).;
8. 4. Sort the result by lenth ;
9. 5. Produce a result in CSV format named result.csv:
10.
11.                    File1,3
12.                    File2,4
13.                    ...
14.                    FileN,3
15. =end
16.
17. require 'csv'
18. result = Hash.new
19. Dir.glob(ARGV[0] + "*").each do |file1|
20.   short1 = file1.split("/").last
21.   ext1 = short1.split(".").last
22.   names1 = short1.split(".").first
23.   digit1 = file1.split("_").last
24.   if (ext1 != "rb") or (digit1.to_i.to_s != digit1) or (short1.scan("_").count != 2)
25.           result[short1] = short1.length
26.   end
27. end
28. Dir.glob(ARGV[1] + "*").each do |file2|
29.   short2 = file2.split("/").last
30.   ext2 = short2.split(".").last
31.   names2 = short2.split(".").first
32.   digit2 = file2.split("_").last
```

```
33.   if (ext2 != "rb") or (digit2.to_i.to_s != digit) or (short2.scan("_").count != 2)
34.           result[short2] = short2.length
35.   end
36. end
37. result.sort_by{|k, v| v}
38. CSV.open("result.csv", "w") do |csv|
39.   result.each do |p|
40.           csv << p
41.   end
42. end
```

There are three mistakes in Ivelin Slavchev's program. The first one is on **line 23** and **line 32**. By splitting the digit the valuable *digit1* will be the number and the extension. The easiest way to fix this problem is to split by "." after splitting by "_". We should add ".split(".").first"

digit1 = file1.split("_").last.split(".").first
digit2 = file2.split("_").last.split(".").first

The second mistake in this program is on **line 33**. On **line 32** the variable is named "digit2", but on **line 33** the same variable is named only "digit".

if (ext2 != "rb") or (digit2.to_i.to_s != digit2) or (short2.scan("_").count != 2)

The third mistake is on **line 37**. We need to add a variable which to save the sorted result.

result = result.sort_by{|k, v| v}
Another way to fix the mistake is to sort the Hash on **line 39**.

result.*sort_by{|k, v| v}*.each do |p|
Rank: 3 of 5

Dimitar Nestorov №7

```
1.  #Develop a program named FirstName_LastName_ClassNumber_0d5526.rb
2.  #Dimitar_Nestorov_7_0d5526.rb
3.  #1. you are given an argument for a folder with files;
4.  #1.1 if there are other arguments they should be discarded
5.  #2. file names in this folder are in the form First_Last_digits.rb;
6.  #3. find all the students that have 10 letters in their first name;
7.  #4. Sort the result by Last Name DESC.
8.  #5. Produce a result in CSV format named result.csv:
9.  #
10. # FirstName1,LastName1
```

```
11. # FirstName2,LastName2
12. # ...
13. # FirstNameN,LastNameN
14. require 'csv'
15. def is_numeric(o)
16. true if Integer(o) rescue false
17. end
18. array = []
19. count = 0
20. Dir.glob(ARGV[0] + "*.rb") do |file|
21.   name = file.split("/").last.split(".").first.split("_")
22.           name[0] = name[0].to_s
23.           name[0] = name[0].capitalize
24.           name[1] = name[1].to_s
25.           name[1] = name[1].capitalize
26.   if name.size == 3 && is_numeric(name[2])
27.         if name[1].length == 10
28.                   array[count] = []
29.                   array[count][0] = name[0].to_s
30.                   array[count][1] = " #{name[1].to_s}"
31.                   count += 1
32.           end
33.   end
34. end
35. array = array.sort_by {|el| -el[1]}
36.       CSV.open("result.csv", "w") do |csv|
37.           array.uniq.each do |e|
38.             csv << e
39.         end
40.       end
```

The mistakes in **Dimitar Nestorov's** program are on:

line 27 where the program checks if **SECOND** name's length is 10. According to the task the program must check if **FIRST** name's length is 10. To fix this mistake we have to change

*if name**[1]**.length == 10* with

*if name**[0]**.length == 10*

line 35 Wrong syntax

*array = array.sort_by {|el| **-**el[1]}* must be

array = array.sort_by {|el| el[1]}

line 35 Where the program sort the result by Last Name ASC .According to the task the program must sort the result by Last Name DESC so fix it by adding **.reverse**

*array = array.sort_by {|el| el[1]}**.reverse***

Rank: 3 of 5

Momchil Angelov №18

```
=begin
Develop a program named FirstName_LastName_ClassNumber_d8aa65.rb

1. you are given two arguments for a folders with files;
1.1 If there are other arguments they should be discarded;
2. Find all the files from both folders that are not in the format
FirsrName_LastName_digits.rb. If there are duplicates the file must be written only once.
2.1 If two files are of the same length those files should be sorted in ASC order;
3. Calculate the length of their names (including extensions).;
4. Sort the result by length ;
5. Produce a result in CSV format named result.csv:

                File1,3
                File2,4
                ...
                FileN,3
=end
require 'csv'

arr1=Array.new
arr2=Array.new
arr3=Array.new
a = ARGV[0]
b = ARGV[1]
i=0
Dir.glob(a + "/*.rb") do |my_text_file1|
  short= my_text_file1.split('/').last
  length1 = short.length
  shorter= short.split('.').first.split('_')
  first_name=shorter[0]
  last_name=shorter[1]
  digits=shorter[2].to_i


  if !first_name || !last_name || digits=0
        next
  else
        arr1 << ["#{short}" "#{length1}"]
  end
end
Dir.glob(b + "/*.rb") do |my_text_file2|
```

```
41.
42.   short2= my_text_file2.split('/').last
43.   length2 = short2.length
44.   shorter2= short.split('.').first.split('_')
45.   first_name2=shorter2[0]
46.   last_name2=shorter2[1]
47.   digits2=shorter2[2].to_i
48.
49.
50.   if !first_name2 || !last_name2 || digits2=0
51.           next
52.   else
53.           arr2 << ["#{short2}","#{length2}"]
54.   end
55. end
56.
57.   arr3 = arr1 & arr2
58.
59.
60.
61.   arr3 = arr3.sort_by  {|el|
62.            el[1]
63.   }
64.
65.
66.
67.           CSV.open("result.csv", "w") do |csv|
68.
69. arr3.each do |element|
70. csv << element
71. end
72.
73. end
```

The first problem in Momchil Angelov's program is on line 34 and line 50.

```
if !first_name || !last_name || digits=0
```

This is if statement so it is wrong to write digit = 0, it has to be digit == 0. The same for **line 50**. Also according to the task the program must find all the files from both folders that are NOT in the format FirsrName_LastName_digits.rb. So we should write the file name only if the if statement return 1. This is how it has to be:

```
if !first_name || !last_name || digits == 0
```

```
            arr1 << ["#{length1}","#{short}"]
    end
```

And the program must check if the extension is .rb. The same for the if statement on **line 50**.
On **line 53**: arr2 is useless. It will be much easier if we use the same array arr1. When we correct this mistake we don't need the code from **line 56** to **line 66**.

There is a mistake on line 44 too - wrong named variable. It must be short2, not shot.

```
    shorter2= short2.split('.').first.split('_')
```

The only thing that left to be done is the sorting. This is how it has to look:

```
    CSV.open("result.csv", "w") do |csv|
      arr1.sort.uniq.each do |element|
            csv << [element].flatten.reverse
      end
    end
```

Rank: 3 of 5

Lubomir Yankov №16

```
1.  require 'csv'
2.  def is_numeric(o)
        a.  true if Integer(o) rescue false
3.  end
4.
5.  array = []
6.  count = 0
7.
8.  Dir.glob(ARGV[0] + "*").each do |file|
9.    ch_count = 0
10.   file_name = file.split("/").last.split("")
11.
12.   file_name.each do |ch|
13.
14.           if is_numeric(ch)
15.
16.                   ch_count += 1
17.
18.           end
19.
```

```
20.   end
21.   if ch_count == 9
22.           len = file_name.length
23.           array[count] = []
24.           array[count][0] = file_name
25.           array[count][1] = len/2.round
26.           count += 1
27.   end
28.
29. end
30. array = array.sort_by {|el| el[0]}
31. CSV.open("result.csv", "w") do |csv|
32.
33.   array.each do |element|
34.
35.           csv << element
36.
37.   end
38.
39. end
```

The mistakes in **Lubomir Yankov**'s program is on:
line 21 where the program checks if file name have exactly **9** digits but according to the task the program must check if file name have exactly **7** digits. So we just replace 9 with 7.
line 31 wrong name of CSV file.

Borislav Rusinov №1

```
1.  =begin
2.  Develop a program named FirstName_LastName_ClassNumber_6fb3ad.rb
3.
4.  1. you are given an argument for a folder with files;
5.  1.1 if there are other arguments they should be discarded
6.  2. file names in this folder are in the form First_Last_digits.rb;
7.  3. find all the students that have 10 letters in their first name;
8.  4. Sort the result by Last Name DESC.
9.  5. Produce a result in CSV format named result.csv:
10.
11.           FirstName1,LastName1
12.           FirstName2,LastName2
13.           ...
14.           FirstNameN,LastNameN
15. =end
```

```
16. a=ARGV[0]
17. require 'csv'
18. array=[]
19. Dir.glob("#{a}*.*")  do |my_text_file|
20.    name = my_text_file.split("/").last.split(".").first.split("_")
21.    if name[1]!=nil && name[0].length == 10
22.              array << name[0] + "," + name[1]
23.    end
24. end
25. array.sort!
26. array.reverse!
27. File.open("results.csv", "w") do |csv|
28.    array.each do |arg|
29.    csv.puts(arg)
30.    end
31. end
```

There is two mistakes in Borislav Rusinov's program. The first one, and the most commen, is the wrong CSV file's name. According to the task the name must be "result.csv" but in the program the name is "results.csv".

File.open("result.csv", "w") do |csv|

The second mistake is in the sorting. Acording to the task the program must sort the result by Last Name DESC. There is not a quick solution to the problam.

Rank: 4 of 5

Denis Trenchev 11a class

```
1.  =begin
2.  Develop a program named FirstName_LastName_ClassNumber_b4c3f5.rb
3.
4.  1. you are given two arguments for a folders with files;
5.  1.1 if there are other arguments they should be discarded;
6.  2. file names in this folders are in the form First_Last_digits.rb;
7.  3. find  the students with 5 letters in the first name that are in both folders. A student is in
    both folders if it there is a file with the same First and Last Name. Digits might be different;
8.  4. Sort the result by Last name ;
9.  5. Produce a result in CSV format named result.csv:
10.
11.   LastName1,FirstName1
12.   LastName2,FirstName2
```

```
13.   ...
14.   LastNameN,FirstNameN
15.
16. =end
17.
18. require 'csv'
19.
20. i = 0
21. arr1 = []
22. arr2 = []
23. arr3 = []
24.
25. Dir.glob(ARGV[0]+"*.rb") do |first_folder|
26.   name = first_folder.split('/').last.split('.').first.split('_')
27.
28.   if name.length == 3
29.           if name[1].to_s.length == 5
30.                   arr1[i] = []
31.                   arr[i][0] = name[0]
32.                   arr[i][1] = name[1]
33.                   i+=1
34.           end
35.   end
36. end
37. i = 0
38.
39. Dir.glob(ARGV[1]+"*.rb") do |second_folder|
40.   name = second_folder.split('/').last.split('.').first.split('_')
41.
42.   if name.length == 3
43.           if name[1].to_s.length == 5
44.                   arr1[i] = []
45.                   arr[i][0] = name_1[0]
46.                   arr[i][1] = name_1[1]
47.                   i+=1
48.           end
49.   end
50. end
51. i = 0
52.
53. arr1.each do |compare1|
54.   arr2.each do |compare2|
55.           if compare2 == compare1
56.                   arr3[i] = compare1
```

```
57.                     i+=1
58.             end
59.    end
60. end
61.
62. sort = arr3.sort_by{|asd| asd[1]}
63. CSV.open("students.csv", "w") do |csv|
64.             sort.each do |element|
65.                     csv << element
66.             end
67. end
```

There is a lot of problems in this program.
Let's start with Dir.glob(ARGV[1]+"*.rb") do |second_folder| . Here the problem are on **line 40, 42 and 43**. The variable is with name "name" but on **line 45** and **line 46** the variable that is supposed to be the same is now with the name "name_1". To fix the mistake we have to name the same variables with the same names.
Too many errors and there is not quick solution to the problems.
Rank: 1 of 5

Dimitar Terziev №6

```
1. =begin
2. Dimitar_Terziev_6_88db52.rb
3. Develop a program named FirstName_LastName_ClassNumber_88db52.rb
4. 1. you are given an argument for a folder with files;
5. 1.1 if there are other arguments they should be discarded
6. 2. file names in this folder are in the form First_Last_digits.rb;
7. 3. find all the students that have 5 letters in their second name;
8. 4. Sort the result by Last Name ASC.
9. 5. Produce a result in CSV format named result.csv:
10. FirstName1,LastName1
11. FirstName2,LastName2
12. ...
13. FirstNameN,LastNameN
14. =end
15. require 'csv'
16. arr = []
17.   Dir.glob("#{ARGV[0]}*.rb*"){|file|
18.           file_str = file.split('/').last
19.             if(file_str=~/\A[a-zA-Z]+\_[a-zA-Z]+\_\d+\.rb\z/ && file_str.split('_')[1].size ==
    5)
20.                         arr.push("#{file_str.split('_')[1]} #{file_str.split('_').first}")
21.             end
22. }
23. CSV.open('result.csv','w'){|csv|
24.   arr.uniq.sort.each{|el|
25.           csv << "#{el.split(' ').last} #{el.split(' ').first}".split(' ')
26.   }
27. }
```

The only mistake in **Dimitar Terziev**'s program is on line 19 where he checks if file name contain only letters from A-Z and a-z. This is not correct according to the task.
Rank: 4 of 5

Ivo Valchev №11

```
1. =begin
2. Develop a program named FirstName_LastName_ClassNumber_6c8bd9.rb
3.
4. 1. you are given two arguments for a folders with files;
```

```
5.  1.1 if there are other arguments they should be discarded;
6.  2. file names in this folders are in the form First_Last_digits.rb;
7.  3. find  the students with 5 letters in the first name that are in both folders. A student is in
    both folders if it there is a file with the same First and Last Name. Digits might be different;
8.  4. Sort the result by Last name ;
9.  5. Produce a result in CSV format named result.csv:
10.
11.   LastName1,FirstName1
12.   LastName2,FirstName2
13.   ...
14.   LastNameN,FirstNameN
15. =end
16.
17. hash_fold1={}
18. hash_fold2={}
19.
20. Dir.glob("#{ARGV[0]}*.*")  do |file|
21.             name = file.split("/").last.split(".").first.split("_")
22.             isNum = Integer(name[2]) rescue nil
23.             if name[0] and name[1] and name[0].length == 5 and !isNum!=nil
    hash_fold1.include?(name[0])
24.                     hash_fold1["#{name[1]}"] = "#{name[0]}"
25.             end
26. end
27. Dir.glob("#{ARGV[1]}*.*") do |file|
28.             name = file.split("/").last.split(".").first.split("_")
29.             isNum = Integer(name[2]) rescue nil
30.             if name[0] and name[1] and name[0].length == 5 and !isNum!=nil
    and!hash_fold2.include?(name[0])
31.                     hash_fold2["#{name[1]}"] = "#{name[0]}"
32.             end
33. end
34. File.open("result.csv", "w") do |csv|
35.   hash_fold1.sort.map do |key, value|
36.             if (hash_fold1[key]==hash_fold2[key])
37.                     csv.puts("#{key},#{value}")
38.             end
39.   end
40. end
```

Too many errors.

Rank: 2 of 5

Kristina Pironkova №15

```
1.  =begin
2.  Develop a program named FirstName_LastName_ClassNumber_890ba0.rb
3.
4.  1. you are given an argument for a folder with files;
5.  1.1 if there are other arguments they should be discarded
6.  2. file names in this folder are in the form First_Last_digits.rb;
7.  3. find all the students that have 10 letters in their first name;
8.  4. Sort the result by Last Name DESC.
9.  5. Produce a result in CSV format named result.csv:
10.
11.         FirstName1,LastName1
12.         FirstName2,LastName2
13.         ...
14.         FirstNameN,LastNameN
15. =end
16.
17. require 'csv'
18. results=Hash.new
19. Directory = ARGV[0]
20. Dir.glob("#{Directory}/*.rb") do |file_name|
21.
22.   first_name = file_name.split("/").last.split("_").first.capitalize
23.   last_name=file_name.split("/").last.split("_",2).last.split("_").first.capitalize
24.
25.         if first_name.length == 10
26.                 results["#{last_name}"] ="#{first_name}"
27.         end
28. end
29.
30. CSV.open("results.csv", "w") do |csv|
31.   results.sort.each do |first,last|
32.         csv << [last,first]
33.   end
34. end
```

There is only one mistake in **Kristina Pironkova**'s program. The problem is in her CSV file's name.
The name should be "result.csv" according the task but it is "results.csv".
To correct that mistake we have to remove the "s" from the CSV file name.

Rank: 5 of 5

Nikola Marinov №20

```
=begin
Nikola_Marinov_20_add57e.rb
1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names excluding extension. If there are duplicates the file must be written only once.;
3. Calculate the length of their names (including extensions) divided by 2 rounded to the smallest number;
4. Sort the result by File name ;
5. Produce a result in CSV format named result.csv:
File1,3
File2,4
...
FileN,3
=end
requre 'csv'
def is_numeric(o)
  true if Integer(o) rescue false
end
array=[]
count=0
Dir.glob(ARGV[0] + "/**/*.*").each do |file|
  full_name=file.split("/").last
  name = file.split("/").last.split(".").first_split("_")
        if name.lenght != 3 && !is_numeric(name[2])
                array(count) = []
                array(count) [0]=full_name
                array(count)[1]= full_name.to_s.lenght
                count += 1
        end
end
Dir.glob(ARGV[0] + "/**/*.*").each do |file|
  full_name=file.split("/").last
  name = file.split("/").last.split(".").first_split("_")
        if name.lenght != 3 && !is_numeric(name[2])
                array(count) = []
                array(count) [0]=full_name
                array(count)[1]= full_name.to_s.lenght
```

```
37.                     count += 1
38.             end
39. end
40. array = array.sort_by{|el| el|0|}
41. CSV.open("task.csv",w) do |csv|
42.   array=uniq.each do |element|
43.             csv << element
44.   end
45. end
```

In **Nikola Marinov's** program there are a lot of syntax mistakes. Let's start with:
line 14 requre 'csv' must be requ**i**re 'csv
line 22 name = file.split("/").last.split(".").**first_split**("_") must be
name = file.split("/").last.split(".").**first.split**("_"). Same thing in **line 32**.
line 23 if name.**lenght** != 3 && !is_numeric(name[2]) must be if name.**length** != 3 && !is_numeric(name[2]). Same in **line 33**.
line 24,25,26,34,35,36 have similar syntax mistakes. array**(**count**)** must be array**[**count**].**Array elements must be write with [ ] brackets not with ( ) brackets.
line 40 array = array.sort_by{|el| el**|0|**} must be array = array.sort_by{|el| el**[0]**}
line 41 CSV.open("task.csv",w) do |csv| wrong named CSV file. Must be result.csv. Wrong syntax CSV.open("task.csv",w) must be CSV.open("task.csv",**"w"**)

There is not quick solution of other problems in this program.

Rank: 2 of 5

Tihomir Lidanski №27

```
1.  #Develop a program named FirstName_LastName_ClassNumber_dafd44.rb
2.
3.  #1. you are given two arguments for a folders with files;
4.  #1.1 if there are other arguments they should be discarded;
5.  #2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names
    excluding extension. If there are duplicates the file must be written only once.;
6.  #3. Calculate the length of their names (including extensions) divided by 2 rounded to the
    smalles number;
7.  #4. Sort the result by File name ;
8.  #5. Produce a result in CSV format named result.csv:
9.
10. #                   File1,3
```

```
11. #                   File2,4
12. #                   ...
13. #                   FileN,3
14.
15. require 'csv'
16.
17. Dir.glob(ARGV[0] + "*.") do |file|
18.    name = file.split ("/")last.split(".")
19.
20.
21. Dir.glob(ARGV[1] + "*.") do |file|
22.
23.
24.
25.
26. puts name.length % 2.round()
27.
28.
29.
30. end
31. end
32.
33. CSV.open("result.csv", "w") do |csv|
34.
35.
36.
37. end
```

There are a lot of problems in Tihomir Lidanski's program.
First of all the program is not checking if the files from both folders have exactly 7 digits from 0 to 9 in their names excluding extension.
There are a lot of missing information in the program and is not a quick solution.

Rank: 1 of 5

Marian Belchev №17

```
1. =begin
2. Develop a program named FirstName_LastName_ClassNumber_ad26e0.rb
3.
4. 1. you are given two arguments for a folders with files;
5. 1.1 if there are other arguments they should be discarded;
6. 2. file names in this folders are in the form First_Last_digits.rb;
```

```
3. find  the students that are only in the second folder and not in the first. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by First name ;
5. Produce a result in CSV format named result.csv:

  LastName1,FirstName1
  LastName2,FirstName2
  ...
  LastNameN,FirstNameN
=end

require 'csv'

hash1 = Hash.new
hash2 = Hash.new

Dir.glob("#{ARGV[0]}*_*_*.rb") do |file1|
  Dir.glob("#{ARGV[1]}*_*_*.rb") do |file2|
          firstName1    = file1.split("/").last.split("_").first
          lastName1     = file1.split("/").last.split("_", 2).last.split("_").first
          number1     = file1.split("_").last.split(".").first

          firstName2    = file2.split("/").last.split("_").first
          lastName2     = file2.split("/").last.split("_", 2).last.split("_").first
          number2     = file2.split("_").last.split(".").first

          hash1[firstName1] = lastName1 + "." + number1
          hash2[firstName2] = lastName2 + "." + number2
  end
end

CSV.open("results.csv", "w") do |csv|
  hash2.sort.each do |key, value|
          if !hash1.has_key?(key) && !hash1.has_value?(value.split(".").first) && !hash1.has_value?(value.split(".").last.to_i)
                  csv << [key,value.gsub('.',"")]
          end
          if hash1.has_key?(key) && !hash1.has_value?(value.split(".").first) && !hash1.has_value?(value.split(".").last.to_i)
                  csv << [key,value.gsub('.',"")]
          end
  end
end
```

There are a few error in Marian Belchev's program. The first problem is on **line 32** and **line 33**. We need if statement for the program to work better:

```
if number1.to_i.to_s == number1
        if !firstName1.empty? && !lastName1.empty?
                hash1[lastName1 + "," + firstName1] = NIL
        end
end

if number2.to_i.to_s == number2
        if !firstName2.empty? && !lastName2.empty?
                hash2[lastName2 + "," + firstName2] = NIL
        end
end
```

The other problem is from **line 39** to **line 45**. This is one way to make the program work:

```
hash2.sort_by{|k,v| k.split(",").last}.each do |key, value|
        if !hash1.has_key?(key)
                firstName = key.split(",").last
                lastName = key.split(",").first
                csv << [lastName,firstName]
        end
end
```

Rank: 5 of 5

Stanislav Valkanov №25

```
1.  #Develop a program named FirstName_LastName_ClassNumber_4482c1.rb
2.
3.  #1. you are given an argument for a folder with files;
4.  #1.1 if there are other arguments they should be discarded
5.  #2. file names in this folder are in the form First_Last_digits.rb;
6.  #3. find all the students that have 5 letters in their second name;
7.  #4. Sort the result by First name DESC.
8.  #5. Produce a result in CSV format named result.csv:
9.
10. #       FirstName1,LastName1
11. #       FirstName2,LastName2
```

```
12. #           ...
13. #           FirstNameN,LastNameN
14.
15. require 'csv'
16. a = Hash.new
17. path = ARGV[0]
18. Dir.glob(path + "**/*.rb") do |my_text_file|
19. short_name = my_text_file.split('/').last.split('.').first
20. name = short_name.split("_")[0]
21. last = short_name.split("_")[1]
22. last.to_s
23. if (last.length == 5)&&(short_name.split("_").size == 3)
24. a["#{name}"] = last
25. end
26. end
27. CSV.open("result.csv", "w") do |csv|
28. Hash[a.sort.reverse].each do |element|
29. csv << element
30. end
31. end
```

In **Stanislav Valkanov's** program mistakes are in:
line 21-22 *last.to_s* don't work so we need to unite that two lines
last = short_name.split("_")[1].to_s
line 28 this line don't sort correctly so we change it to ***a.sort.reverse.each do |element|***
The program is not structured well.

Rank: 3 of 5

Veselin Dechev №11A

```
1.  require 'csv'
2.  result = Hash.new
3.  Dir.glob(ARGV[0] + "*.rb").each do |first|
4.    name1 = first.split("/").last.capitalize
5.    first_name = name1.split("_").first.capitalize
6.    last_name = name1.split("_",2).last.split('_').first.capitalize
7.    Dir.glob(ARGV[1]+"*.rb").each do |second|
8.          name2 = second.split("/").last.capitalize
9.          if (name1 == name2)
10.               result.compare_by_identity
```

```
11.                     result[first_name] = last_name
12.             end
13. end
14. end
15. CSV.open("result.csv", "w") do |csv|
16.   result.sort_by{|k, v| k}.each do |element|
17.             csv << element
18.             end
19.   end
20.
```

According to the task the program must have specific name. Veselin Dechev's program is with wrong name. Also Veselin Dechev did not write the task that was given to him.
On **line 16** the program will sort the students by first name. We can fix that mistake by changing this:
result.sort_by{|k, v| k}.each do |element|
with this:
result.sort_by{|k, v| v}.each do |element|

On **line 4** and **line 8** by splitting the directory only by "/" the valuable will be the file name, not the student name. The program will not work write if there is the same student in both folders, but with different digits. Also there is no need of the *capitalize* there. This problem can be fix by after splitib by "/" to split by "." and to get the first element.

name1 = first.split("/").last.split(".").first
Acording to the task the program must find the students that are **only** in the first folder and **not** in the second. Veselin's program check if the student is in both folders. There is not a quick solution to this problem. One opportunity to fix this problem is with two more hashes. One for the files from the first folder, and one for the second folder. After that we have to compare the names from both files and to write only those that are only in the fisrt file in the CSV file.

Rank: 4 of 5

Stanislav Gospodinov №26

```
1. =begin
2. Develop a program named FirstName_LastName_ClassNumber_b36abb.rb
3.
4. 1. you are given an argument for a folder with files;
5. 1.1 if there are other arguments they should be discarded
```

```
2. file names in this folder are in the form First_Last_digits.rb;
3. find all the students that have 5 letters in their second name;
4. Sort the result by Last Name ASC.
5. Produce a result in CSV format named result.csv:

        FirstName1,LastName1
        FirstName2,LastName2
        ...
        FirstNameN,LastNameN
=end

require 'csv'
hash = Hash.new

Dir.glob("#{ARGV[0]}*.rb") do |file|
  filename = file.split('/').last.split('.').first;
        if filename.split('_').length == 3
                if filename.split('_')[1].length == 5
                        hash[filename.split('_')[0]] = filename.split('_')[1]
                end
        end
end

hash = Hash[hash.sort_by{|k, v| v}]

CSV.open("results.csv", "w") do |csv|
  hash.each do |key, value|
        csv << [key, value].flatten
  end
end

```

In **Stanislav Gospodinov's** program mistakes are in:
line 29 - 32. In **line 31** csv file name isn't correct. Sorting isn't work well so we change it to **hash.sort.each do |key, value|** on **line 32**

Rank: 5 of 5

Kalin Marinov № 12

```
1. #==begin
```

```
2.  #Develop a program named FirstName_LastName_ClassNumber_bce70c.rb
3.  #
4.  #1. you are given an argument for a folder with files;
5.  #1.1 if there are other arguments they should be discarded
6.  #2. file names in this folder are in the form First_Last_digits.rb;
7.  #3. find all the students that have 5 letters in their second name;
8.  #4. Sort the result by First name DESC.
9.  #5. Produce a result in CSV format named result.csv:
10. #
11. #       FirstName1,LastName1
12. #       FirstName2,LastName2
13. #       ...
14. #       FirstNameN,LastNameN
15. #==end
16.
17. require 'csv'
18.
19. hash = Hash.new
20.
21. Dir.glob("#{ ARGV[0] }/*") do |name|
22.   name = name.split("/").last
23.   short_name = name.split('_')[1]
24.   if short_name.length == 5
25.       hash[name] = short_name
26.   end
27. end
28.
29. CSV.open("result.csv", "w") do |csv|
30.   hash = hash.sort_by { |key, value| value }.reverse
31.   hash.each |key| do
32.       csv << key
33.   end
34. end
```

The mistakes in **Kalin Marinov's** program are in:
line 22-25. As key the program saves in CSV file the full name of the files from the folder and as value last name of the students.
On **line 23** written like that ***(short_name = name.split('_')[1])*** short_name = nil. So we add **.to_s** *short_name = name.split('_')[1]**.to_s.***
But ,we don't know why, even with this correction sometimes short_name take last name of the student and file extension.
To fix it , we change code from **line 22** to **line 25**

```
Dir.glob("#{ ARGV[0] }/*") do |name|
    name1 = name.split(/\//).last.split(/_/).first#Take first name
```

```
        name = name.split("/").last
        short_name = name.split(/\//).last.split(/_/, 2).last.split("_").first.split(".").first #Take last name without extension
                if short_name.length == 5
                        hash[name1] = short_name #Save in hash only FIRST name as a key
                end
  end
```

There aren't any checks if file is in a format First_Last_digits.rb. Because of that information in the CSV file is wrong.

Rank: 3 of 5

Simeon Shopkin №23

```
1.  =begin
2.  Develop a program named FirstName_LastName_ClassNumber_56a835.rb
3.
4.  1. you are given two arguments for a folders with files;
5.  1.1 if there are other arguments they should be discarded;
6.  2. Find all the files from both folders that are not in the format
    FirsrName_LastName_digit.rb. If there are duplicates the file must be written only once. If
    two files are of the same length those files should be sorted in ASC order;
7.  3. Calculate the length of their names (including extensions).;
8.  4. Sort the result by length ;
9.  5. Produce a result in CSV format named result.csv:
10.
11.                 File1,3
12.                 File2,4
13.                 ...
14.                 FileN,3
15.
16. =end
17.
18. require 'csv'
19.
20. arr = Array.new
21.   Dir.glob(ARGV[0]+"/*.rb") do |first_files|
22.         Dir.glob(ARGV[1]+"/*.rb") do |second_files|
23.                 first_files = first_files.split("/").last.split(".").first.split("_")
24.                 if first_files.size != 3
25.                         if first_files != second_files
```

```
26.                                         print_count = first_files.split("/").last.split(".").first
27.                                         p = print_count.size.to_s
28.                                         print =
    first_files[0].capitalize+"_"+first_files[1].capitalize+"_"+first_files[2]+","+p
29.                                         arr.push(print)
30.                             end
31.                     end
32.             end
33.     end
34.
35.     CSV.open("result.csv","w") do |csv|
36.             arr.sort.each do |element|
37.                     csv << [element]
38.             end
39.     end
```

There are a lot of mistakes in Simeon Shopkin's program.
Firstly, on **line 21** and **line 22** we have to remove ".rb", because according to the task the program must find the files that are not in the format FirstName_LastName_digit.rb. If the file is with another extension the program must find it.
Secondly, on **line 23** by splitting the valuable first_files and make it equal to the same valuable it will change. On **line 28** the program is trying to split the already splitted valuable. So we need different valuable.
On **line 24** the program is checking only if there is three words in the name but there are other thing we must check for. For example the digit can be something else then a number, or the extension to be different from .rb. So we need more condition.
Also, on **line 27** the program compare the valuables first_files and second_files. But those valuables are the directory to the files from the different folders, so they will never be the same. We need another valuables that to be equal to the file names. And after that to compare them, not the directory.
We need Hash, because according to the task the program must sort the result by length.
There isn't a quick solution to the problems.

Rank: 2 of 5

Petko Bozhinov № 21

```
1.  # Develop a program named FirstName_LastName_ClassNumber_954dc6.rb
2.
3.  # 1. you are given two arguments for a folders with files;
4.  # 1.1 if there are other arguments they should be discarded;
5.  # 2. file names in this folders are in the form First_Last_digits.rb;
```

```
# 3. find  the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
# 4. Sort the result by Last name ;
# 5. Produce a result in CSV format named result.csv:

#    LastName1,FirstName1
#    LastName2,FirstName2
#    ...
#    LastNameN,FirstNameN

require 'csv'

class String
 def numeric?
            Float(self) != nil rescue false
 end
end

output = Array.new
i = 0
Dir.glob(ARGV[0] + "/*") do |file|
   file = file.split('/').last.split('.').first.split('_')
   Dir.glob(ARGV[1] + "/*") do |file2|
            file2 = file2.split('/').last.split('.').first.split('_')
            if "#{file[0]} #{file[1]}" == "#{file2[0]} #{file2[1]}"
                        if file[2].numeric?
                                    if file[0].to_s.length == 5
                                                output[i] = Array.new
                                                output[i][0] = file[0]
                                                output[i][1] = file[1]
                                                i+=1
                                    end
                        end
            end
   end
end

output = output.sort_by{ |element| element[1]}
CSV.open("result.csv", "w") do |csv|
   output.each do |pusher|
            csv << pusher
   end
end
```

In ***Petko Bozhinov's*** program the mistake is in **line 30 (if file[2].numeric?)** where he missed to add **.to_s**
if file[2]**.to_s**.numeric?

Also if we want the information in the CSV file to look better we can add .uniq:

```
CSV.open("result.csv", "w") do |csv|
        output.uniq.each do |pusher|
        csv << pusher
        end
end
```

The program isn't sorting result by first name so we delete **line 42** and add to **line 44 .sort**

```
CSV.open("result.csv", "w") do |csv|
        output.uniq.sort.each do |pusher|
        csv << pusher
        end
end
```

Rank: 5 of 5

11b class

Borislav Stratev №

```
1.  #Develop a program named FirstName_LastName_ClassNumber_a65be5.rb
2.
3.  #1. you are given two arguments for a folders with files;
4.  #1.1 if there are other arguments they should be discarded;
5.  #2. file names in this folders are in the form First_Last_digits.rb;
6.  #3. find  the students that are only in the first folder and not in the second. A student is in
    both folders if it there is a file with the same First and Last Name. Digits might be different;
7.  #4. Sort the result by Last name ;
8.  #5. Produce a result in CSV format named result.csv:
9.
10. #    LastName1,FirstName1
11. #    LastName2,FirstName2
12. #    ...
13. #    LastNameN,FirstNameN
14.
15. require 'csv'
16. a = Array.new
17. h = Hash.new
18. Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
19.   Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|
20.
21.           file_name_1 = dir_file_name_1.split(/\//).last.to_s
22.           file_name_2 = dir_file_name_2.split(/\//).last.to_s
23.
24.           if(file_name_1 != file_name_2)
25.                   file_name = file_name_1
26.                   digit = file_name.split(/_/).last.split(/\./).first.to_s
27.                   first_name = file_name.split(/_/).first.to_s
28.                   full_first_name = first_name + digit
29.                   full_first_name = full_first_name.to_s
30.                   tmp = file_name.split("#{first_name}_")
31.                   full_last_name = tmp.last.split(/_/).first.to_s + digit
32.                   full_last_name = full_last_name.to_s
33.                   h[full_last_name] = full_first_name
34.
35.           end
36.   end
37. end
```

```
38.
39. CSV.open("results.csv", "w") do |csv|
40.   a = h.sort
41.   a.each do |element|
42.           csv << element
43.   end
44. end
```

The CSV file's name is wrong. According to the task it must be "result.csv" but in the program it is "results.csv".
According to the task a student is in both folders if it there is a file with the same First and Last Name. Digits might be different. So on **lines 21 and 22** we have to split the file names and by ".", Like that we will get only the names without the digit.
There is error on **lines 28 and 31**. We have to remove *"+ digit".*

Rank: 3 of 5

David Georgiev №12

```
1.  #Develop a program named FirstName_LastName_ClassNumber_1eea4f.rb
2.
3.  #1. you are given an argument for a folder with files;
4.  #1.1 if there are other arguments they should be discarded
5.  #2. file names in this folder are in the form First_Last_digits.rb;
6.  #3. find all the students that have 5 letters in their second name;
7.  #4. Sort the result by Last Name ASC.
8.  #5. Produce a result in CSV format named result.csv:
9.
10. #       FirstName1,LastName1
11. #       FirstName2,LastName2
12. #       ...
13. #       FirstNameN,LastNameN
14.
15.  require 'csv'
16.  students_names = []
17.  Dir.glob("#{ARGV[0]}/**/*.rb") do |current_file|
18.
19.  name = current_file.split('/').last.split(/_/)
20.  if name[1].length == 5
21.          if not students_names.include?(["#{name[1]}", "#{name[0]}"]) then
22.                  students_names << (["#{name[1]}", "#{name[0]}"])
```

```
23.             end
24.     end
25.     end
26.     CSV.open("result.csv", "w") do |csv|
27.             students_names.sort.each do |last, first|
28.                     csv << ["#{first}",  "#{last}"]
29.             end
30.     end
```

There is no problem in David Georgiev's program.
Rank: 5 of 5

Iliqn Germanov №17

```
1.  =begin
2.    Develop a program named FirstName_LastName_ClassNumber_f8b0d9.rb
3.
4.  1. you are given two arguments for a folders with files;
5.  1.1 if there are other arguments they should be discarded;
6.  2. file names in this folders are in the form First_Last_digits.rb
7.  3. find the students that are only in the first folder and not in the second. A student is in
    both folders if it there is a file with the same First and Last Name. Digits might be different;
8.  4. Sort the result by Last name ;
9.  5. Produce a result in CSV format named result.csv:
10.
11.   LastName1,FirstName1
12.   LastName2,FirstName2
13.   ...
14.   LastNameN,FirstNameN
15. =end
16.
17. require 'csv'
18. results = Hash.new
19. results.compare_by_identity
20. def is_number(str)
21.   str[/[0-9]+/] == str
22. end
23. Dir.glob("#{ARGV[0]}/*.rb") do |path1|
24.   filename1 = path1.split(/\//).last
25.   if filename1.count("_") == 2
26.           firstname1 = filename1.split("_").first
27.           lastname1 = filename1.split("_")[1]
```

```
28.             digit1 = filename1.split("_")[2].split(".").first
29.             if is_number(digit1)
30.                     flag = 0
31.                     Dir.glob("#{ARGV[1]}/*.rb") do |path2|
32.                             filename2 = path2.split(/\//).last
33.                             if filename2.count("_") == 2
34.                                     digit2 = filename2.split("_")[2].split(".").first
35.                                     if is_number(digit2)
36.                                             name1 = firstname1 + lastname1
37.                                             name2 = filename2.split("_").first +
    filename2.split("_")[1]
38.                                             if name1 == name2
39.                                                     flag = 1
40.                                                     break
41.                                             end
42.                                     end
43.                             end
44.                     end
45.                     if flag == 0
46.                             results[lastname1] = firstname1
47.                     end
48.             end
49.   end
50. end
51.
52. CSV.open("result.csv", "w") do |csv|
53.   results.sort_by{|key, val| key}.each do |el|
54.             csv << el
55.   end
56. end
```

There is no problem in David Georgiev's program.
Rank: 5 of 5

Lili Kokalova №22

```
1.  =begin
2.  Develop a program named FirstName_LastName_ClassNumber_e0ea9c.rb
3.
4.  1. you are given two arguments for a folders with files;
5.  1.1 if there are other arguments they should be discarded;
```

```
2. file names in this folders are in the form First_Last_digits.rb;
3. find  the students that are only in the second folder and not in the first. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by First name ;
5. Produce a result in CSV format named result.csv:

  LastName1,FirstName1
  LastName2,FirstName2
  ...
  LastNameN,FirstNameN
=end

require 'csv'
student = Array.new
student1 = Array.new

Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
  file_name = file_name1.split("/").last
  first_name = file_name.split("/").last.split("_").first
  p first_name
  last_name = file_name.split("/").last.split("_",2).last.split("_").first
  #task = file_name.split("_").last.split(".").first
  student << ["#{first_name}", "#{last_name}"]
end

Dir.glob(ARGV[1]+"/**/*.*").each do  |file_name1|
  file_name = file_name1.split("/").last
  first_name = file_name.split("/").last.split("_").first
  p first_name
  last_name = file_name.split("/").last.split("_",2).last.split("_").first
  #task = file_name.split("_").last.split(".").first
  student1 << ["#{first_name}", "#{last_name}"]
end

CSV.open("result.csv", "w") do |csv|
  student.each do |fn, ln|
        student1.each do |fn1, ln1|
              if fn != fn1
                    if ln != ln1
                          csv << ["#{fn1}", "#{ln1}"]
                    end
              end
        end
  end
```

```
49. end
```

The mistake in **Lili Kokalova**'s program is in lines 42 and 43. The problem is that always in some moment the file names form the folders will be different, so it will always go in the loop.
To fix this mistake we have to add a variable on line 38 that to be equal false, for example c=false.
On line 42 to 46 to write:

```
if fn == fn1
        if ln == ln1
                c = true
        end
end
```

So when the program finds same files from the folders the variable c will become true. Now between **lines 47** and **48** we must add loop that will write in the CSV file only if the variable c is flase.

```
if c == false
        csv << ["#{fn}", "#{ln}"]
end
c = false
```

We need to make the variable c equal false in case that it had become true on **line 44**.

Rank: 4 of 5

Nikola Mihailov №25

```
1. #Develop a program named FirstName_LastName_ClassNumber_f70059.rb
2.
3. #1. you are given two arguments for a folders with files;
4. #1.1 if there are other arguments they should be discarded;
5. #2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names
   excluding extension. If there are duplicates the file must be written only once.;
6. #3. Calculate the length of their names (including extensions) divided by 2 rounded to the
   smallest number;
7. #4. Sort the result by File name ;
8. #5. Produce a result in CSV format named result.csv:
9.
10. #                   File1,3
11. #                   File2,4
```

```
12. #                   ...
13. #                   FileN,3
14.
15. require 'csv'
16. hash = Hash.new
17. count = 0
18.   Dir.glob(ARGV[0] + "/*.rb") do |file|
19.
20.           first = file.split(/\//).last
21.           puts first
22.
23.           #for (i = 0;i < first.length;i+=1)
24.           size = first.length
25.           i = 0
26.           first.each do |element|
                i.
27.                   c = first[i].chr
28.                   if element == 0 || element == 1 || element == 2 || element == 3 || element
    == 4 || element == 5 || element == 6 || element == 7 || element == 8 || element == 9
29.                   count +=1
30.
31.                   end
32.           end
33.           puts count
34.   end
35.
36.   Dir.glob(ARGV[1] +"/*.rb") do |secFile|
37.           sec = secFile.split(/\//).last
38.           #puts sec
39.
40.   end
41.
42.   CSV.open("result.csv", "w") do |csv|
43.           hash.sort_by{|key,val| key}.each do |element|
44.           csv << element
45.           end
46.   end
```

To make Nikola Mihailov's program work we should change the code form line 21 to line 33 with:

```
if first.count("0-9") == 7
        hash[first] = (first.length / 2).ceil
end
```

This three line code checks if there is 7 digits from 0 to 9 in the file name and calculate the length of their names (including extensions) divided by 2 rounded to the smalles number.

From **line 37 to line 39** we should write similar thing:

```
first = secFile.split(/\//).last
if first.count("0-9") == 7
        hash[first] = (first.length / 2).ceil
end
```

Rank: 2 of 5

Stanislav Ilieav

```
1.  #Develop a program named FirstName_LastName_ClassNumber_627d43.r#
2.  #
3.  #1. you are given two arguments for a folders with files;
4.  #1.1 if there are other arguments they should be discarded;
5.  #2. file names in this folders are in the form First_Last_digits.rb;
6.  #3. find the students that are only in the first folder and not in the second. A student is in
    both folders if it there is a file with the same First and Last #Name. Digits might be different;
7.  #4. Sort the result by Last name ;
8.  #5. Produce a result in CSV format named result.csv:
9.  #
10. #   LastName1,FirstName1
11. #   LastName2,FirstName2
12. #   ...
13. #   LastNameN,FirstNameN
14.
15.
16.
17. require 'csv'
18. name_array = Array.new()
19. name_array2 = Array.new()
20. support_array = Array.new()
21. support_array2 = Array.new()
22. i = 0
23. dir1 = ARGV[0]
24. dir2= ARGV[1]
25.
26. Dir.glob("#{dir1}/*.*") do |file|
27.    name_array[i] = file.split(/\//).last
```

```
28.    i += 1
29. end
30. count = i
31. i = 0
32. Dir.glob("#{dir2}/*.*") do |file2|
33.    name_array2[i] = file2.split(/\//).last
34.    i += 1
35. end
36. i = 0
37. for check in i..count
38.    if name_array[check] != name_array2[check]
39.                        support_array[i] = name_array[check]
40.                        support_array2[i] = name_array2[check]
41.                        i += 1
42.                        puts support_array
43.                        puts support_array2
44.                        CSV.open("result.csv", "w") do |csv|
45.                                support_array.each do |element|
46.                                        csv << [element]
47.                                end
48.                        end
49.                        CSV.open("result.csv", "w") do |csv|
50.                                support_array2.each do |element2|
51.                                                csv << [element2]
52.                                end
53.                        end
54.    end
55. end
```

There are too many error in Stanislav Ilieav's program.
Let's start with the fact that the expected results should be:

```
LastName1,FirstName1
LastName2,FirstName2
...
LastNameN,FirstNameN
```

But Stanislav'sprogram split the diredctory only by "/" that will return the file names, not the student names.
The same for the second dir glob.
Also according to the task the program must find the students that are only in the first folder and not in the second. We can solve this problem by checking every file form the first folder with the files from the second folder and to repeat that until the program checks all the files form the first

folder. If there is not the same student in the both folders the program to write it in the CSV file after sorting them by last name (“sort_by{|k, v| v}”).

Rank: 3 of 5

Valentin Varbanov №4

```
=begin

Develop a program named FirstName_LastName_ClassNumber_041472.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb;
3. find the students that are only in the first folder and not in the second. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by Last name ;
5. Produce a result in CSV format named result.csv:

  LastName1,FirstName1
  LastName2,FirstName2
  ...
  LastNameN,FirstNameN


=end

students_first_dir = Array.new
students_second_dir = Array.new

for i in 0..1

  directory = ARGV[i]
  if ARGV[i].split(//).last(1).to_s == "/"
        directory += "**/*.rb"
  else
        directory += "/**/*.rb"
  end

  Dir.glob(directory).each do |dir|
        student = dir.split(/\//)
        if i == 0
```

```
35.                     students_first_dir.push(student)
36.             else
37.                     students_second_dir.push(student)
38.             end
39.     end
40. end
41.
42. studentcsv = Array.new
43.
44. students_first_dir.each do |std|
45.     match = 0
46.     students_second_dir.each do |std2|
47.             name = std.last.split(/_/)
48.
49.             name2 = std2.last.split(/_/)
50.             for i in 0..1
51.                     if name[i] == name2[i]
52.                             match = 1
53.                     end
54.             end
55.
56.     end
57.     studentcsv.push(name[1], name[2])
58. end
59.
60. CSV.open("result.csv", "w") do |csv|
61.     studentcsv.each do |string|
62.             csv << string
63.     end
64. end
```

The pirst problem in Valentin Varbanov's program is on line 47. This (name = std.last.split(/_/)) must be between lines 45 and 46 because of line 57. Otherwise we can't use the valuable name.
The next problem is thaht according the task the program must write first the last name and after thet the first name, and to sort by last name. So between line 56 and 57 we need to write:

```
name[1] = name.first
name[0] = std.last.split("_", 2).last.split("_").first
```

Those two lines will make name[1] equal the first name and name[0] equal the last name.
This: studentcsv.push(name[1], name[2]) is wrong too. First of all name[2] will be the extension, but we don't need it. So it has to be studentcsv.push(name[0], name[1]). But like that the names won't be on one line. This is wrong according to the task. That is why we need to write

studentcsv.push(name[0..1]) so the students names to be on one line when the program put them in the CSV file.
Also we need if statement.

```
if (match == 0)
        studentcsv.push(name[0..1])
end
```

Now the program will put the students that are only in the first folder.

There is a few more errors.
One of them is on **line 61**. According to the task the program must sort the students. So we need the function .sort.
And the other error is on **line 62**.

```
CSV.open("result.csv", "w") do |csv|
  studentcsv.uniq.sort.each do |string|
        puts string
        csv << [string].flatten
  end
end
```

Rank: 4 of 5

Veselina Kolova №8

```
1. =begin
2. Develop a program named FirstName_LastName_ClassNumber_65630e.rb
3.
4. 1. you are given an argument for a folder with files;
5. 1.1 if there are other arguments they should be discarded
6. 2. file names in this folder are in the form First_Last_digits.rb;
7. 3. find all the students that have 5 letters in their second name;
8. 4. Sort the result by First name DESC.
9. 5. Produce a result in CSV format named result.csv:
10.
11.         FirstName1,LastName1
12.         FirstName2,LastName2
13.         ...
14.         FirstNameN,LastNameN
15. =end
16.
```

```
17. require 'csv'
18.
19. people = Hash.new
20.
21. Dir.glob("#{ARGV[0]}/**/*.*").each do |text_file|
22.
23.   if File.extname(text_file) text_file.include?(".rb") &&
    text_file.split(/_/).last.split(/\./).first.to_i.is_a Integer then
24.           if (text_file.split("/").last.split("_").length == 3) then
25.                   text_file = text_file.split("/").last
26.                   if (text_file.split("_")[1].length == 5) then
27.                   people[text_file.split("_")[1]] = text_file.split("_")[0]
28.                   end
29.           end
30.   end
31. end
32.
33. people = Hash[people.sort_by{|k,v| k}.reverse]
34.
35. CSV.open("result.csv","w") do |csv|
36.           people.each do |element|
37.           csv << element
38.           end
39. end
```

Without the if statement on **line 23** the program works perfectly.
Rank: 5 of 5

Vladimir Yordanov №9

```
1.  #Develop a program named FirstName_LastName_ClassNumber_4bbed0.rb
2.
3.  #1. you are given an argument for a folder with files;
4.  #1.1 if there are other arguments they should be discarded
5.  #2. file names in this folder are in the form First_Last_digits.rb;
6.  #3. find all the students that have 5 letters in their second name;
7.  #4. Sort the result by Last Name ASC.

8.  #5. Produce a result in CSV format named result.csv:
9.
10. #           FirstName1,LastName1
```

```
11. #           FirstName2,LastName2
12. #           ...
13. #           FirstNameN,LastNameN
14.
15.
16.
17. names = Hash.new
18. Dir.glob (ARGV[0] + "*.rb") do |file|
19.   if (ARGV[1] == true)
20.           ARGV[1] == false
21.   end
22.
23.   slice = file.split("/").last
24.   first_name = slice.split('_')[0]
25.   second_name = slice.split('_')[1]
26.   if (second_name.length == 5)
27.           #print first_name
28.           #puts second_name
29.           names[first_name] = second_name
30.   end
31.
32. end
33.
34. names = names.sort
35. puts names
36.
37. require 'csv'
38. CSV.open("results.csv", "w") do |csv|
39.   names.to_a.each do |element|
40.           csv << element
41.   end
42. end
```

There are only two mistakes in Vladimir Yordanov's program.
The first one is the wrong name of the CSV file. According the task the name must be "result.csv" but in the program the name is "results.csv".
The other mistake is in the sorting. According the task the results must be sort by last name. To fix this mistake **line 34** should look like that:
names = names.sort_by{|k, v| v}

instead of:
names = names.sort

Rank: 5 of 5