



**Technological School “Electronic Systems”
associated with
Technical University Sofia**

Report

<LIST>

Lili Kokalova, 11b, №22

Stefan Radev, 11b, №27

Date: 25.10.2014

Learning from errors

“What are the errors and how could these errors be avoided in the first place?”

I. How many errors are there?

Officially there are 34 committed tests and only 5 of them are correct therefore there are around 85.3% incorrect tests. Eventually we discovered that there are 2 more correct tests which reduces the percent from around 85.3% to around 79.4% .

II. How many correct results are there?

Officially there are 34 committed tests and only 5 of them are correct therefore there are around 14.7% correct tests. Eventually we discovered that there are 2 more correct tests which raise the percent from around 14.7% to around 20.6 %. There are 8 students with correct and nearly correct results distributed in two categories:

1. The students whose programs work although their CSV files' names are not correct are:

Kristina_Pironkova_15_890ba0.rb

Borislav_Rusinov_2_6fb3ad.rb

Stefan_Iliev_28_d77aee.rb

Stanislav_Gospodinov_26_b36abb.rb

2. The other students with correct programs are:

Iliyan_Germanov_17_f8b0d9.rb

David_Georgiev_12_1eea4f.rb

III.Are there common errors?

The most common error is that the students did not read their task correctly and they named their csv files "*results.csv*" instead of "*result.csv*". Another common mistake is that the students did not sort the names in the correct way. Other students in their hurry have mistaken their variables and the places they use them. Another mistake is that a lot of the students don't check for correct file name format. A lot of the errors could not be fixed.

IV.Are there categories of errors?

The categories of errors are:

- Wrong names of CSV files
- Wrong sorting
- Wrong output
- Wrong syntax
- Wrongly understood task

V.How could errors like this be avoided in first place?

The students could avoid errors by reading carefully what they are needed to do in their task. Another thing they can do is understand the Ruby programming language syntax and use it correctly for preventing errors like writing the method "*.length*" correctly. Other thing is by using the appropriate brackets in their rightful places for example: for arrays they should be the square ones "[]" not the round ones "()". Last but not least errors could be avoided if the student have more time to do the test.

Appendixes

1. Borislav_Rusinov_2_6fb3ad.rb

Develop a program named `FirstName_LastName_ClassNumber_6fb3ad.rb`

1. you are given an argument for a folder with files;
 - 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 10 letters in their first name;
4. Sort the result by Last Name DESC.
5. Produce a result in CSV format named `result.csv`:

```

        FirstName1,LastName1
        FirstName2,LastName2
        ...
        FirstNameN,LastNameN
a=ARGV[0]
require 'csv'
array=[]
Dir.glob("#{a}*.rb") do |my_text_file|
    name = my_text_file.split("/").last.split(".").first.split("_")
    if name[1]!=nil && name[0].length==10
        array << name[0] + "," + name[1]
    end
end
array.sort!
array.reverse!
File.open("results.csv", "w") do |csv| #results.csv should be result.csv
    array.each do |arg|
        csv.puts(arg)
    end
end
```

The only problem here is that the CSV file should be named *result.csv*. Despite that the program is working without a problem.

Rank: 4

2. Denis_Trenchev_4_b4c3f5.rb

Develop a program named FirstName_LastName_ClassNumber_b4c3f5.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 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'

i = 0
arr1 = []
arr2 = []
arr3 = []

Dir.glob(ARGV[0]+"*.rb") do |first_folder|
  name = first_folder.split('/').last.split('.').first.split('_')

  if name.length == 3
    if name[1].to_s.length == 5
      arr1[i] = []
      arr[i][0] = name[0]  #arr[i] not defined; it should be arr1[i]
      arr[i][1] = name[1]  #arr[i] not defined; it should be arr1[i]
      i+=1
    end
  end
end

i = 0

Dir.glob(ARGV[1]+"*.rb") do |second_folder|
  name = second_folder.split('/').last.split('.').first.split('_')  #name defined in first glob; it
  should be name_1

  if name.length == 3
    if name[1].to_s.length == 5
      arr1[i] = []
      arr[i][0] = name_1[0]  #arr[i] not defined; it should be arr2[i]
      arr[i][1] = name_1[1]  #arr[i] not defined; it should be arr2[i]
      i+=1
    end
  end
end
```

```

        end
    end
    i = 0

    arr1.each do |compare1|
        arr2.each do |compare2|
            if compare2 == compare1
                arr3[i] = compare1 # arr3[i] << compare1
                i+=1
            end
        end
    end
end

sort = arr3.sort_by{|asd| asd[1]}
CSV.open("students.csv", "w") do |csv| #it should not be students.csv but result.csv
    sort.each do |element|
        csv << element
    end
end

```

This program can't be fixed quickly.

Rank: 2

3. Dimitar_Nestorov_7_0d5526.rb

Develop a program named FirstName_LastName_ClassNumber_0d5526.rb

1. you are given an argument for a folder with files;
 - 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 10 letters in their first name;
4. Sort the result by Last Name DESC.
5. Produce a result in CSV format named result.csv:

```

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

```

```

require 'csv'
def is_numeric(o)
    true if Integer(o) rescue false
end
array = []

```

```

count = 0
Dir.glob(ARGV[0] + "*.rb") do |file|
  name = file.split("/").last.split(".").first.split("_")

  name[0] = name[0].to_s
  name[0] = name[0].capitalize

  name[1] = name[1].to_s
  name[1] = name[1].capitalize

  if name.size == 3 && is_numeric(name[2])
    if name[1].length == 10 #it should be name[0] not name[1] because we're
searching for the first name
      array[count] = []
      array[count][0] = name[0].to_s
      array[count][1] = "#{name[1].to_s}"
      count += 1
    end
  end
end
array = array.sort_by {|e| -e[1]} #without the dash in front of e[1]
CSV.open("result.csv", "w") do |csv|

  array.uniq.each do |e|

    csv << e

  end

end

```

The program is working despite the mistakes and is readable.

Rank: 3

4. Dimitar_Terziev_6_88db52.rb

Develop a program named FirstName_LastName_ClassNumber_88db52.rb

1. you are given an argument for a folder with files;
 - 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

```

```

require 'csv'
arr = []
Dir.glob("#{ARGV[0]}*.rb*"){|file| #there should be a backslash in the **
  file_str = file.split('/').last
  if(file_str =~ /\A[a-zA-Z]+\_[a-zA-Z]+\_\d+\.rb\z/ && file_str.split('_')[1].size == 5)
    arr.push("#{file_str.split('_')[1]} #{file_str.split('_').first}")
  end
}
CSV.open('result.csv','w'){|csv|
  arr.uniq.sort.each{|el|
    csv << "#{el.split(' ').last} #{el.split(' ').first}".split(' ')
  }
}

```

The program cannot be fixed quickly and it is not readable.

Rank: 1

5.Ivelin_Slavchev_10_835552.rb

```

require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*").each do |file1|
  short1 = file1.split("/").last
  ext1 = short1.split(".").last
  names1 = short1.split(".").first
  digit1 = file1.split("_").last
  if (ext1 != "rb") or (digit1.to_i.to_s != digit1) or (short1.scan("_").count != 2)
    result[short1] = short1.length
  end
end
Dir.glob(ARGV[1] + "*").each do |file2|
  short2 = file2.split("/").last
  ext2 = short2.split(".").last
  names2 = short2.split(".").first
  digit2 = file2.split("_").last
  if (ext2 != "rb") or (digit2.to_i.to_s != digit2) or (short2.scan("_").count != 2) #instead of
digit it should be digit2
    result[short2] = short2.length
  end
end

```



```

        end
    end
    result.sort_by{|k, v| v}
    CSV.open("result.csv", "w") do |csv|
        result.each do |p|
            csv << p
        end
    end
end

```

Despite the mistake the program does not work properly because the output is wrong and it cannot be fixed quickly.

Rank: 2

6.Ivo_Valchev_11_6c8bd9.rb

Develop a program named FirstName_LastName_ClassNumber_6c8bd9.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 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

```

```

hash_fold1={}
hash_fold2={}

```

```

Dir.glob("#{ARGV[0]}*.rb") do |file|
    name = file.split("/").last.split(".").first.split("_")
    isNum = Integer(name[2]) rescue nil
    if name[0] and name[1] and name[0].length == 5 and !isNum!=nil and
hash_fold1.include?(name[0]) #there must be exclamation mark(!) in front of hash_fold1
        hash_fold1["#{name[1]}"] = "#{name[0]}"
    end
end
end
Dir.glob("#{ARGV[1]}*.rb") do |file|
    name = file.split("/").last.split(".").first.split("_")
    isNum = Integer(name[2]) rescue nil

```

```

        if name[0] and name[1] and name[0].length == 5 and !isNum!=nil
and!hash_fold2.include?(name[0])
            hash_fold2["#{name[1]}"] = "#{name[0]}"
        end
    end
end
File.open("result.csv", "w") do |csv|
    hash_fold1.sort.map do |key, value|
        if (hash_fold1[key]==hash_fold2[key])
            csv.puts("#{key},#{value}")
        end
    end
end
end

```

Despite the error the program is working but it is not readable.

Rank: 3

7.Kalin_Marinov_14_12_bce70c.rb **#wrong name**

Develop a program named FirstName_LastName_ClassNumber_bce70c.rb

1. you are given an argument for a folder with files;
- 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 First name DESC.
5. Produce a result in CSV format named result.csv:

```

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

```

```
require 'csv'
```

```
hash = Hash.new
```

```

Dir.glob("#{ ARGV[0] }/*") do |name|
    name = name.split("/").last
    short_name = name.split('_')[1]
    if short_name.length == 5
        hash[name] = short_name
    end
end
end

```

```

CSV.open("result.csv", "w") do |csv|
    hash = hash.sort_by { |key, value| value }.reverse
    hash.each |key| do    #|key| must be after do

```

```

        csv << key
      end
    end
  end
end

```

The program does not sort the names correctly but it is readable.

Rank: 3

8. Kristina_Pironkova_15_890ba0.rb

Develop a program named FirstName_LastName_ClassNumber_890ba0.rb

1. you are given an argument for a folder with files;
 - 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 10 letters in their first name;
4. Sort the result by Last Name DESC.
5. Produce a result in CSV format named result.csv:

```

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

```

```

require 'csv'
results=Hash.new
Directory = ARGV[0]
Dir.glob("#{Directory}/*.rb") do |file_name|

  first_name = file_name.split("/").last.split("_").first.capitalize
  last_name=file_name.split("/").last.split("_",2).last.split("_").first.capitalize

  if first_name.length == 10

    results["#{last_name}"] = "#{first_name}"
  end
end

CSV.open("results.csv", "w") do |csv|  #results.csv should be result.csv
  results.sort.each do |first,last|

    csv << [last,first]

  end
end

```

end

Despite the mistake the program is working and it is readable.

Rank: 4

9. Lubomir_Yankov_16_650c0b.rb

Develop a program named FirstName_LastName_ClassNumber_650c0b.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

```
require 'csv'
def is_numeric(o)
  true if Integer(o) rescue false
end

array = []
count = 0

Dir.glob(ARGV[0] + "*").each do |file|
  ch_count = 0
  file_name = file.split("/").last.split(".")

  file_name.each do |ch|

    if is_numeric(ch)

      ch_count += 1

    end

  end

  end

  if ch_count == 9
```

```

        len = file_name.length
        array[count] = []
        array[count][0] = file_name
        array[count][1] = len/2.round
        count += 1
    end

end

array = array.sort_by {|el| el[0]}
CSV.open("results.csv", "w") do |csv|    #results.csv should be result.csv

    array.each do |element|

        csv << element

    end

end
end

```

Despite the mistake the program cannot be fixed quickly.

Rank: 1

10. Marian_Belchev_17_ad26e0.rb

Develop a program named FirstName_LastName_ClassNumber_ad26e0.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 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

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|  #results.csv should be result.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
end

```

The program has wrong output. It prints every name from the second folder and cannot be fixed quickly.

Rank: 2

11. Momchil_Angelov_18_d8aa65.rb

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 FirstName_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

```
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|

  short2= my_text_file2.split('/').last
  length2 = short2.length
  shorter2= short2.split('.').first.split('_')
  first_name2=shorter2[0]
  last_name2=shorter2[1]
  digits2=shorter2[2].to_i

  if !first_name2 || !last_name2 || digits2=0
    next
  else
    arr2 << ["#{short2}", "#{length2}"]
  end
end

arr3 = arr1 & arr2
arr3 = arr3.sort_by {|el|
```

```

        el[1]
    }

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

arr3.each do |element|
  csv << element
end

end

```

The program cannot be fixed quickly.

Rank: 1

12. 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

```

```

require 'csv'           #it should be require
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("_")  #it should be first.split("_") not first_split

  if name.length != 3 && !is_numeric(name[2]) #it should be length not lenght
    array(count) = [] #it should be array[count]
  end
end

```



```

array(count) [0]=full_name      #it should be array[count]
array(count)[1]= full_name.to_s.lenght      #it should be array[count]; it should be
length not lenght
count += 1

end
end

Dir.glob(ARGV[0] + "**/*.*").each do |file|

full_name=file.split("/").last
name = file.split("/").last.split(".").first_split("_")      #it should be first.split("_") not first_split

if name.lenght != 3 && !is_numeric(name[2])      #it should be length not lenght
array(count) = []      #it should be array[count]
array(count) [0]=full_name      #it should array[count]
array(count)[1]= full_name.to_s.lenght      #it should array[count]; it should be length not
length
count += 1
end
end
array = array.sort_by{|el| el[0]}      #el[0] should be el[0]

CSV.open("task.csv",w) do |csv|      #task.csv should be result.csv; w should be in quotes "w"
array=uniq.each do |element|      #it should be array.uniq.each not array=uniq.each
csv << element
end
end

```

This code is not understandable and has a lot of mistakes. The program cannot be fixed quickly.

Rank: 1

13. Petko_Bozhinov_21_954dc6.rb

Develop a program named FirstName_LastName_ClassNumber_954dc6.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 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| #instead of one variable in || there should be two
    csv << pusher
  end
end
end

```

When the mistake is fixed the program is working but it is not very readable.

Rank: 3

14. Radoslav_Kostadinov_22_772118.rb

Develop a program named FirstName_LastName_ClassNumber_772118.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 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

```

```

require 'csv'
file1 = Hash.new
file2 = Hash.new

path1 = ARGV[0]
path2 = ARGV[1]

Dir.glob("#{path1}*.rb") do |my_text_file|
  s = my_text_file.split(/\//).last.capitalize
  first_name = my_text_file.split("/").last.split("_").first
  last_name = my_text_file.split("/").last.split("_",2).last.split("_").first
  if s.count('_') == 2 and !((first_name == "" || first_name == " ") || (last_name == "" || last_name == " "))
    file1[first_name] = last_name
  end
end

Dir.glob("#{path2}*.rb") do |my_text_file|
  s = my_text_file.split(/\//).last.capitalize
  first_name = my_text_file.split("/").last.split("_").first
  last_name = my_text_file.split("/").last.split("_",2).last.split("_").first

  if s.count('_') == 2 and !((first_name == "" || first_name == " ") || (last_name == "" || last_name == " "))
    file2[first_name] = last_name
  end
end

CSV.open("result.csv", "w") do |csv|
  file1.sort.each do |first_name, last_name|
    file2.sort.each do |first_name1, last_name1|

```

```

        if first_name1 == first_name and last_name1 == last_name
          begin      #instead of begin and end it should be
          end        #file2.delete(first_name1)
        else
          csv << [last_name1, first_name1] #wrong writing in csv
        end
      end
    end
  end
  #right writing in csv
  #file2.sort.each do |first_name,last_name|
  #      csv << [last_name, first_name]
  #end
end

```

There are some mistakes but despite them it is working and is readable.
Rank: 2

15. Simeon_Shopkin_23_56a835.rb

Develop a program named FirstName_LastName_ClassNumber_56a835.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 FirstName_LastName_digit.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;
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

```

```
require 'csv'
```

```

arr = Array.new
Dir.glob(ARGV[0]+"/*.rb") do |first_files|
  Dir.glob(ARGV[1]+"/*.rb") do |second_files|
    first_files = first_files.split("/").last.split(".").first.split("_")
    if first_files.size != 3
      if first_files != second_files

```

```

        print_count = first_files.split("/").last.split(".").first #it should be
without .split(".").first
        p = print_count.size.to_s
        print=
first_files[0].capitalize+"_"+first_files[1].capitalize+"_"+first_files[2]+", "+p #
        arr.push(print)
    end
end
end
end

CSV.open("result.csv","w") do |csv|
    arr.sort.each do |element| #sorting is wrong it should be by two variables
        csv << [element]
    end
end
end

```

The program is not working despite the fixed mistakes and cannot be fixed quickly.

Rank: 1

16. Stanislav_Gospodinov_26_b36abb.rb

Develop a program named FirstName_LastName_ClassNumber_b36abb.rb

1. you are given an argument for a folder with files;
 - 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

```

```

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|  #it should be result.csv instead of results.csv
    hash.each do |key, value|
        csv << [key, value].flatten
    end
end
end

```

Despite the mistake it is working without a problem and it is readable.

Rank: 4

17. Stanislav_Valkanov_25_4482c1.rb

Develop a program named FirstName_LastName_ClassNumber_4482c1.rb

1. you are given an argument for a folder with files;
 - 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 First name DESC.
5. Produce a result in CSV format named result.csv:

```

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

```

```

require 'csv'
a = Hash.new
path = ARGV[0]
Dir.glob(path + "**/*.rb") do |my_text_file|
    short_name = my_text_file.split('/').last.split('.').first
    name = short_name.split("_")[0]
    last = short_name.split("_")[1]
    last.to_s
    if (last.length == 5)&&(short_name.split("_").size == 3) #method .length should be
outside and before the if
        a["#{name}"] = last
    end
end
end

```

```

CSV.open("result.csv", "w") do |csv|
  Hash[a.sort.reverse].each do |element|
    csv << element
  end
end
end

```

The program is working despite the mistake.

Rank: 4

18. Tihomir_Lidanski_27_dafd44.rb

Develop a program named FirstName_LastName_ClassNumber_dafd44.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

```

```
require 'csv'
```

```
Dir.glob(ARGV[0] + "/*.") do |file|
  name = file.split("/").last.split(".")

```

```
Dir.glob(ARGV[1] + "/*.") do |file|
```

```
puts name.length % 2.round()
```

```

end
end

```

```
CSV.open("result.csv", "w") do |csv|
```

```
End
```

The program is not working. It cannot be fixed.

Rank: 1

19. Veselin_Dechev_11A2_5f1c22.rb #wrong name

```
require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*.rb").each do |first|
  name1 = first.split("/").last.capitalize
  first_name = name1.split("_").first.capitalize
  last_name = name1.split("_",2).last.split('_').first.capitalize
  Dir.glob(ARGV[1]+"*.rb").each do |second|
    name2 = second.split("/").last.capitalize
    if (name1 == name2)
      result.compare_by_identity
      result[first_name] = last_name
    end
  end
end
end
CSV.open("result.csv", "w") do |csv|
  result.sort_by{|k, v| k}.each do |element|
    csv << element
  end
end
```

The program has wrong output and it cannot be fixed quickly.

Rank: 1

20. David_Georgiev_12_1eea4f.rb

Develop a program named FirstName_LastName_ClassNumber_1eea4f.rb

1. you are given an argument for a folder with files;
 - 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
```

```
require 'csv'
students_names = []
```



```

Dir.glob("#{ARGV[0]}/**/*.rb") do |current_file|

  name = current_file.split('/').last.split(/_/)
  if name[1].length == 5
    if not students_names.include?("#{name[1]}", "#{name[0]}") then
      students_names << (["#{name[1]}", "#{name[0]}"])
    end
  end
end
end
CSV.open("result.csv", "w") do |csv|
  students_names.sort.each do |last, first|
    csv << ["#{first}", "#{last}"]
  end
end
end

```

The program is working without problems.

Rank: 5

21. Iliyan_Germanov_17_f8b0d9.rb

Develop a program named FirstName_LastName_ClassNumber_f8b0d9.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

```

```

require 'csv'
results = Hash.new
results.compare_by_identity
def is_number(str)
  str[/[0-9]+/] == str
end
Dir.glob("#{ARGV[0]}/*.rb") do |path1|
  filename1 = path1.split(/\/\//).last
  if filename1.count("_") == 2
    firstname1 = filename1.split("_").first

```

```

        lastname1 = filename1.split("_")[1]
        digit1 = filename1.split("_")[2].split(".").first
        if is_number(digit1)
            flag = 0
            Dir.glob("#{ARGV[1]}/*.rb") do |path2|
                filename2 = path2.split(/\/\//).last
                if filename2.count("_") == 2
                    digit2 = filename2.split("_")[2].split(".").first
                    if is_number(digit2)
                        name1 = firstname1 + lastname1
                        name2 = filename2.split("_").first +
filename2.split("_")[1]

                        if name1 == name2
                            flag = 1
                            break
                        end
                    end
                end
            end
        end
        if flag == 0
            results[lastname1] = firstname1
        end
    end
end

end

end

CSV.open("result.csv", "w") do |csv|
    results.sort_by{|key, val| key}.each do |el|
        csv << el
    end
end
end

```

The program is working without problems.

Rank: 5

22. Vladimir_Yordanov_9_4bbed0.rb

Develop a program named FirstName_LastName_ClassNumber_4bbed0.rb

1. you are given an argument for a folder with files;
 - 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
```

```
names = Hash.new
Dir.glob (ARGV[0] + "*.rb") do |file|
  if (ARGV[1] == true)
    ARGV[1] == false
  end

  slice = file.split("/").last
  first_name = slice.split('_')[0]
  second_name = slice.split('_')[1]
  if (second_name.length == 5)
    #print first_name
    #puts second_name
    names[first_name] = second_name
  end
end

names = names.sort #it should be names = names.sort_by{|f, l| l}
puts names

require 'csv'
CSV.open("results.csv", "w") do |csv| #it should be result.csv not results.csv
  names.to_a.each do |element|
    csv << element
  end
end
```

When the mistakes are fixed the program is working without problems.

Rank: 4

23. Stefan_Iliev_28_d77aee.rb

Develop a program named FirstName_LastName_ClassNumber_d77aee.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_digit.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;

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
```

```
require 'csv'
```

```
first_folder = ARGV.shift
second_folder = ARGV.shift || "err"
names_hash = Hash.new
```

```
Dir.glob(first_folder+"/*.*").each do |text_file|
  text_file = text_file.split("/").last
  if (text_file.split("_").length == 3) then
    first_name = text_file.split("_")[0]
    second_name = text_file.split("_")[1]
    diggit = text_file.split("_")[2].split(/\./).first
    if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
    if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
    if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
  else
    names_hash[text_file] = text_file.length
  end
end
```

```
if second_folder != "err"
  Dir.glob(second_folder+"/*.*").each do |text_file|
    text_file = text_file.split("/").last
    if (text_file.split("_").length == 3) then
      first_name = text_file.split("_")[0]
      second_name = text_file.split("_")[1]
      diggit = text_file.split("_")[2].split(/\./).first
      if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length
    end

    if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
    if (second_name =~ /\d/) then names_hash[text_file] = text_file.length
  end

  else
    names_hash[text_file] = text_file.length
  end
end

end
```

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

```
puts names_hash
```

```
CSV.open("results.csv","w") do |csv|
  names_hash.each do |element|
    csv << element
  end
end
```

#it should be result.csv not results.csv

Despite the mistake the program is working.

Rank: 4

24. Valentin_Varbanov_4_041472.rb

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
```

There should be *require 'csv'*

```
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
    students_first_dir.push(student)
  else
    students_second_dir.push(student)
  end
end
end

studentcsv = Array.new

students_first_dir.each do |std|
  match = 0
  students_second_dir.each do |std2|
    name = std.last.split(/_/)

    name2 = std2.last.split(/_/)
    for i in 0..1
      if name[i] == name2[i]
        match = 1
      end
    end
  end

  end
  studentcsv.push(name[1], name[2])
end

CSV.open("result.csv", "w") do |csv|
  studentcsv.each do |string|
    csv << string
  end
end

```

The program cannot be fixed quickly.

Rank: 1

25. Nikolay_Mihailov_25_f70059.rb

Develop a program named FirstName_LastName_ClassNumber_f70059.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
```

```
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
end
```

```
  Dir.glob(directory).each do |dir|
    student = dir.split(/\/)
    if i == 0
      students_first_dir.push(student)
    else
      students_second_dir.push(student)
    end
  end
end
```

```
end
```

```
studentcsv = Array.new
```

```
students_first_dir.each do |std|
  match = 0
  students_second_dir.each do |std2|
    name = std.last.split(/_/)

    name2 = std2.last.split(/_/)
    for i in 0..1
      if name[i] == name2[i]
        match = 1
      end
    end
  end
end
```

```

        end
        studentcsv.push(name[1], name[2])
    end

    CSV.open("result.csv", "w") do |csv|
        studentcsv.each do |string|
            csv << string
        end
    end
end

```

The program cannot be fixed quickly.

Rank: 1

26. Borislav_Stratev_2_a65be5.rb

Develop a program named FirstName_LastName_ClassNumber_a65be5.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

```

```

require 'csv'
a = Array.new
h = Hash.new

Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
    Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|

        file_name_1 = dir_file_name_1.split(/\/\//).last.to_s
        file_name_2 = dir_file_name_2.split(/\/\//).last.to_s

        if(file_name_1 != file_name_2)
            file_name = file_name_1
            digit = file_name.split(/_/\//).last.split(/\.\/\//).first.to_s
            first_name = file_name.split(/_/\//).first.to_s
            full_first_name = first_name + digit

```



```

        full_first_name = full_first_name.to_s
        tmp = file_name.split("#{first_name}_")
        full_last_name = tmp.last.split(/_/).first.to_s + digit
        full_last_name = full_last_name.to_s
        h[full_last_name] = full_first_name
    end
end
end

CSV.open("results.csv", "w") do |csv|    #it should be result.csv not results.csv
    a = h.sort
    a.each do |element|
        csv << element
    end
end
end

```

The program cannot be fixed quickly. It prints all the names from the first folder with the task's digit behind it.

Rank: 2

27. Veselina_Kolova_8_65630e.rb

Develop a program named FirstName_LastName_ClassNumber_65630e.rb

1. you are given an argument for a folder with files;
 - 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 First name DESC.
5. Produce a result in CSV format named result.csv:

```

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

require 'csv'

people = Hash.new

Dir.glob("#{ARGV[0]}/**/*.*").each do |text_file|

    if File.extname(text_file) text_file.include?(".rb") &&
text_file.split(/_/).last.split(/\.\/).first.to_i.is_a Integer then    #this should be deleted
        if (text_file.split("/").last.split("_").length == 3) then

```

```

        text_file = text_file.split("/").last    #it should be text_file =
text_file.split("/").last.split("_")
        if (text_file.split("_")[1].length == 5) then    #it should be without .split("_")
            people[text_file.split("_")[1]] = text_file.split("_")[0]    #it should be
people[text_file[0]] = text_file[1]
        end
    end
end
end
end

```

```

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

```

```

CSV.open("result.csv","w") do |csv|
  people.each do |element|
    csv << element
  end
end

```

With the fixed mistakes the program is working.

Rank: 2

28. Stanislav_Iliev_26_627d43.rb

Develop a program named FirstName_LastName_ClassNumber_627d43.r#

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

```

```

require 'csv'
name_array = Array.new()
name_array2 = Array.new()
support_array = Array.new()
support_array2 = Array.new()
i = 0
dir1 = ARGV[0]
dir2= ARGV[1]

```

```

Dir.glob("#{dir1}/*.rb") do |file|
  name_array[i] = file.split(/\/\//).last
  i += 1
end
count = i
i = 0
Dir.glob("#{dir2}/*.rb") do |file2|
  name_array2[i] = file2.split(/\/\//).last
  i += 1
end
i = 0
for check in i..count
  if name_array[check] != name_array2[check]
    support_array[i] = name_array[check]
    support_array2[i] = name_array2[check]
    i += 1
    puts support_array
    puts support_array2
    CSV.open("result.csv", "w") do |csv|
      support_array.each do |element|
        csv << [element]
      end
    end
    CSV.open("result.csv", "w") do |csv|
      support_array2.each do |element2|
        csv << [element2]
      end
    end
  end
end
end
end

```

This program cannot be fixed quickly.

Rank: 1

29. Lili_Kokalova_22_e0ea9c.rb

Develop a program named FirstName_LastName_ClassNumber_e0ea9c.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 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

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
```

#the problem is in the writing in the csv

```
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
end
```

#The solution is:

c = false

```
CSV.open("result.csv", "w") do |csv|
  student1.uniq.each do |fn, ln|
    student.uniq.each do |fn1, ln1|
      if fn == fn1
        if ln == ln1
          c = true
        end
      end
    end
  end
end
```

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

The program is readable but it is not working.

Rank:2