



Презентация за грешките и възможните решения от контролното

Technical School Electronic Systems
October.2014

Team:
RubyDevelopers

From: Hristiyan Velyakov, Veselin Dechev

Report

Common mistakes

1. The most common mistake is having the output **CSV** file wrong named.
2. Another mistake is undefined method in few programs
- 3 .Another frequent mistake is the wrong sorting of the output data
4. There are have few programs with syntax errors

Common solutions:

1. In many of the programs the errors are caused by misreading, or misunderstanding. So the most common solution is **paying enough attention**
2. Even though, there are programs that require more time to be corrected. A very frequent solution is checking the splits. If all the splits are working, the chance of mistake is reduced by a half.

Borislav Rusinov

Task

=begin

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

=end

Program

```
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|
  array.each do |arg|
    csv.puts(arg)
  end
end
```

His program which works correctly .The problem is only csv name.

Program rank 5

Denis Trenchev

Task

=begin

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

=end

Program

```
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][1] = name[1]
```

```
      i+=1
```

```
    end
```

```
  end
```

```

end
i = 0

Dir.glob(ARGV[1]+"*.rb") do |second_folder|
  name = second_folder.split('/').last.split('.').first.split('_')

  if name.length == 3
    if name[1].to_s.length == 5
      arr1[i] = []
      arr[i][0] = name_1[0]
      arr[i][1] = name_1[1]
      i+=1
    end
  end
end
i = 0

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

sort = arr3.sort_by{|asd| asd[1]}
CSV.open("students.csv", "w") do |csv|
  sort.each do |element|
    csv << element
  end
end

```

Denis has program with many wrongs .Few of them are incorrect csv name, undefined array .The program check second name for 5 letters instead first name.

Produce a result in CSV format named result.csv:

```

File.open("students.csv", "w") do |csv|
  find the students with 5 letters in the first name
  if name[1].to_s.length == 5

```

Program rank 3

Dimitar Nestorov

Task

```
#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
```

Program

```
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

      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 {|el| -el[1]}
```

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

```
array.uniq.each do |e|
```

```
csv << e
```

```
end
```

```
end
```

has program which has 2 problems .First of them is program check second name for 10 letters instead first name.Second problems is error by sort.

find all the students that have 10 letters in their first name;

```
if name[1].length == 10
```

```
array = array.sort_by {|el| -el[1]}-error
```

Program rank 4

Kristina Pironkova

Task

```
=begin
```

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
```

```
=end
```

Program

```
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.sort.each do |first,last|

    csv << [last,first]

  end
end
```

Kristina has program which works correctly .The problem is only csv name.

Produce a result in CSV format named result.csv:

```
File.open("results.csv", "w") do |csv|
```

Program rank 5

Lubomir Yankov

Program

```
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
```

```
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|
```

```
  array.each do |element|
```

```
    csv << element
```

```
  end
```

```
end
```

his program check for 9 digits in name instead 7 and wrong csv name .

if ch_count == 9 must be if ch_count == 7

Program rank 5

Petko Bozhinov
Task

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

Program

```
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
end

```

this program works but in some cases doesn't work. It doesn't work when we have wrong name like this Petko_Bozhinov.rb We can solve this problem . If we put one "if" which check for name size.
Program rank 5

Stanislav Gospodinov

Task

```
=begin
```

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
=end

```

Program

```

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

```

```
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
```

Stanislav has program which works correctly. The problem is only csv name.

Produce a result in CSV format named result.csv:

```
File.open("results.csv", "w") do |csv|
```

Program rank: 5

Veselin Dechev

Program

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

his program has few problems.1-st of them is split string wrong .It takes name including extensions.Other problem is program compare names from both folders wrong.It compare for equal names instead output names which are only on 1-st folder.

Program rank:3

Tihomir Lidanski

Task

has program which has a little part of code .There missing big parts from code .

Program rank: 2

#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

Program

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

Stanislav Vulkanov

Task

#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

Program

```
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)
```

```
    a["#{name}"] = last
```

```
  end
```

```

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

```

*his program has little problem. With this row
 If(last.length==5)&&(short_name.split("_").size==3) if we reverse this will
 work correctly
 Program rank:5*

Radoslav Kostadinov

Task

```

=begin
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

```

```

=end

```

Program

```

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
end
else
csv << [last_name1, first_name1]
end
end
end
end
```

*has problem is program compare names from both folders wrong. It compare for **equal** names instead output names which are only on 2-nd folder.*

```
if first_name1 == first_name and last_name1 == last_name-wrong
Program rank :3
```

Dimitar Terziev

Task

His program works properly ,but some parts of code are hard to read
Program rank:4

=begin

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

=end

Program

```
require 'csv'
```

```
arr = []
```

```
Dir.glob("#{ARGV[0]}*.rb*"){|file|
```

```
  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(' ')
```

```
  }
```

```
}
```

```
=
```

Simeon Shopkin

Task

=begin

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

=end

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

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|

csv << [element]

end

end

His program has undefined method for array
Program rank :3

Borislav Stratev

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

LastName + digits , FirstName + digits to: LastName1 ,
FirstName1
- Find the students that are only in the first folder and not in the second. A
student is in both folders if there is a file with the same First and Last Name.
Digits might be different

results.csv to result.csv
Program rank:3

David Georgiev

```
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]}"))

```

Wrong split

Don't sort Last Name по ASC >> Sort the result by Last Name ASC.

Program rank 4

Lili Kokalova

Task

=begin

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

=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
end

```

```

Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
  ..
  ..
  Dir.glob(ARGV[1]+"/**/*.*").each do |file_name1|
    file_name1 is already taken
  end
end

```

Don't output in format

```

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

```

```

csv << ["#{fn1}", "#{ln1}"] Should be in<< ["#{ln1}", "#{fn1}"]
Program rank 4

```

Iliyan Germanov

Don't sort properly

3. Sort the result by Last name ;
4. Program rank 5;

Nikolay Mihailov

#2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names excluding extension.

Don't split properly>> first = file.split(\\).last

Don't have limit to 7 digits>> that have exactly 7 digits

c = first[i].chr

```
if element == 0 || element == 1 || element == 2 || element == 3 || element == 4 ||  
element == 5 || element == 6 || element == 7 || element == 8 || element == 9  
count +=1
```

#3. Calculate the length of their names (including extensions) divided by 2 rounded to the smallest number;

This is missing in the code

Program rank 4

Valentin Varbanov

first folder and not in the second

LastName1,FirstName1

LastName2,FirstName2

...

LastNameN,FirstNameN

Incorrect format

Program rank :4

Stefan Iliev

```
#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
end

```

Wrong split>> that are not in the format FirsrName_LastName_digit.rb

#5. Produce a result in CSV format named result.csv:
results.csv to result.csv
Program rank:5

Stanislav Iliev

Don't split in format>> form First_Last_digits.rb
Don't sort properly>> #4. Sort the result by Last name ;
LastName1,FirstName1
LastName2,FirstName2
...


```
# LastNameN,FirstNameN
```

#5. Produce a result in CSV format named result.csv:

```
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]
```

Write in twice CSV
Program rank :4

Veselina Kolova

Task

=begin

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
=end
```

Program

```
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
    if (text_file.split("/").last.split("_").length == 3) then
      text_file = text_file.split("/").last
      if (text_file.split("_")[1].length == 5) then
        people[text_file.split("_")[1]] = text_file.split("_")[0]
      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

```

Vladimir Yordanov

Task

#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
puts names

require 'csv'
CSV.open("results.csv", "w") do |csv|
  names.to_a.each do |element|
    csv << element
  end
end
end

```

Wrong Sort #4. Sort the result by Last Name ASC.

**#5. Produce a result in CSV format named result.csv: >> results.csv
to result.csv**
Program rank:4

