Иво Алексеев Стратев

Борислав Алексеев Стратев

Technological School “Electronic Systems”

associated with Technical University - Sofia

www.elsys-bg.org

27.10.2014 г.

From 34 evaluated results 25 are wrong and 2 are really wrong wich is 27 total wrong results. This is 79.6% results with errors and 5 out of 34 correct results wich is 14.7% no error results. Acording to our “try corect errors” task there are 2 (5.7%) result that didn’t had any errors at all and we think that they were wrong evaluated.

There are alot of common errors such as: worng csv.file name or inability of using Hashes. Because of that we decided to seperated these errors in categories :

Categorie 1 : Wrong csv.file name.

Categorie 2 : Sorting Hash.map (No knowledge that the RESULT afther sorting Hash is Array.

Category 3 : Inability to work with Arrays in Rubby.

Categorie 4 : Inability to make deferense betwen delimiter and pattern when calling “.split“ method for string class.

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

Categorie 6 : Inability to read task correctly.

Categorie 7 : No task found in the code (Found only in results from A class).

Categorie 8 : Lexical and Language (Rubby) errors in Syntax. (Found only in results from A class).

All these types of errors can be easily avoided by doing next recommendations for each of the categories :

Categorie 1 : Read properly what the name of the output CSV.file should be.

Categorie 2 : Read about what you do not know how to use or using for the first time.

Categorie 3 : Read about what you do not know how to use.

Categorie 4 : Read about what you do not know how to use.

Categorie 5 : Write your homewroks! and write code EVERY day!

Categorie 6 : Learn to read properly.

Categorie 7 : Listen to what the Teacher says!

Categorie 8 : Learn to write properly. Watch for words that you write and are part of the Language (Rubby) that you are using. Changing their apirence in the flow of the text editor you are using. Also learn Language specific reserved words and Symbols!

Appendix for Borislav_Stratev from B class:

#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|
    a = h.sort
    a.each do |element|
        csv << element
    end
end
```

Problem : Errors in following Categories Found:

Categorie 1 : Wrong csv.file name.

Categorie 6 : Inability to read task correctly.

```
a = Array.new
h = Hash.new
Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
    name_1 = dir_file_name_1.split(/\//).last.split(/\./).first.split(/_/)
    first_name_1 = name_1[0]
    last_name_1 = name_1[1]
    check = 0
    Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|
        name_2 = dir_file_name_2.split(/\//).last.split(/\./).first.split(/_/)
        first_name_2 = name_2[0]
        last_name_2 = name_2[1]
```

```
            if(first_name_1 == first_name_2) and (last_name_1 == last_name_2)
                    check = 1
            end

        end
        if(check == 0)
            h[last_name_1] = first_name_1
        end
end

require 'csv'
CSV.open("result.csv", "w") do |csv|
        a = h.sort
        a.each do |element|
            csv << element
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.
        # It is easy to understand, read and work with this program.
Appendix for Denis_Trenchev from A class:

```
=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

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
```

Problem : Errors in following Categories Found:

Categorie 1 : Wrong csv.file name.

Category 3 : Inability to work with Arrays in Rubby.

Categorie 4 : Inability to make deferense betwen delimiter and pattern when calling “.split“ method for string class.

Categorie 6 : Inability to read task correctly.

Categorie 8 : Lexical and Language (Rubby) errors in Syntax. (Found only in results from A class).

```
require 'csv'


arr1 = Hash.new
arr2 = Hash.new
arr3 = Hash.new
sort = Array.new

Dir.glob(ARGV[0]+"/*.rb") do |first_folder|
        name = first_folder.split(/\//).last.split(/\./).first.split(/_/)
                if name[0].to_s.length == 5
                        arr1[name[1]] = name[0]
                end
end

Dir.glob(ARGV[1]+"/*.rb") do |second_folder|
        name2 = second_folder.split(/\//).last.split(/\./).first.split(/_/)
                if name2[0].to_s.length == 5
                        arr1[name2[1]] = name2[0]
                end
end

arr1.each_pair do |k,v|
        arr2.each_pair do |k2,v2|
                if k+v == k2+v2
                        arr3[k] = v
                        i+=1
                end
        end
end

sort = arr3.sort_by{|k,v| v}
```

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

Program is in Rank :1 Where 1 is worst and 5 is best.

        # It is not easy to understand, read and work with this program.

Appendix for Ivelin_Slavchev from A class :

```
=begin
    Develop a program named FirstName_LastName_ClassNumber_835552.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. 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 lenth ;
    5. Produce a result in CSV format named result.csv:

                File1,3
                File2,4
                ...
                FileN,3
=end

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 != digit) or
(short2.scan("_").count != 2)
			result[short2] = short2.length
		end
	end
	result.sort_by{|k, v| v}
	CSV.open("result.csv", "w") do |csv|
		result.each do |p|
			csv << p
		end
	end
```

Problem : Errors in following Categories Found:

Categorie 4 : Inability to make deferense betwen delimiter and pattern when calling “.split“ method for string class.

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

```
require 'csv'
a = Array.new
result = Hash.new
Dir.glob(ARGV[0] + "*").each do |file1|
	short1 = file1.split(/\//).last
	if(/[A-Za-z]+(_)[A-Za-z]+(_)\d+(.rb)/.match(short1))
	else
		result[short1] = short1.length
	end
```

```
end
Dir.glob(ARGV[1] + "*").each do |file2|
        short2 = file2.split(/\//).last
        if(/[A-Za-z]+(_)[A-Za-z]+(_)\d+(.rb)/.match(short2))
        else
                result[short2] = short2.length
        end
end
a = result.sort_by{|k, v| v}
CSV.open("result.csv", "w") do |csv|
        a.each do |p|
                csv << p
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Ivo_Valchev from A class:

=begin
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
```

=end

```
hash_fold1={}
hash_fold2={}

Dir.glob("#{ARGV[0]}*.*")  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 hash_fold1.include?(name[0])
                hash_fold1["#{name[1]}"] = "#{name[0]}"
        end
end
Dir.glob("#{ARGV[1]}*.*") 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
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
```

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

```
hash_fold1= Hash.new
hash_fold2= Hash.new

Dir.glob("#{ARGV[0]}*.*")  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
            hash_fold1["#{name[1]}"] = "#{name[0]}"
        end
end
Dir.glob("#{ARGV[1]}*.*") 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
                    hash_fold2["#{name[1]}"] = "#{name[0]}"
            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
```

Program is in Rank : 3.Where 1 is worst and 5 is best.

```
        # It is medium  to understand, read and  work with this program.
```

Appendix for Kalin_Marinov form A class:

```
#==begin
#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
#==end

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

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

```
                csv << key
            end
        end
```

Problem : Errors in following Categories Found:

Categorie 2 : Sorting Hash.map (No knowledge that the RESULT afther sorting Hash is Array.

```
require 'csv'

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

CSV.open("result.csv", "w") do |csv|
        array = hash.sort.reverse
                array.each do |element|
                csv << element
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

It is easy to understand, read and work with this program.

Appendix for Kristina_Pironkova from A class:

```
=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


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.capitali
ze

        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
```

Problem : Errors in following Categories Found:

        # Categorie 1 : Wrong csv.file name.

        # Categorie 6 : Inability to read task correctly.

```
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.capitali
ze

                if first_name.length == 10

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

end


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

        csv << [last,first]
```

```
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Marian_Belchev from A class:

```
=begin
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
=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
```

Problem : Errors in following Categories Found:

    # Categorie 1 : Wrong csv.file name.

    # Categorie 6 : Inability to read task correctly.

```
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("result.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
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

It is easy to understand, read and work with this program.

Appendix for Dimitar_Nestorov from A class:

```
#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

                        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
```

Problem : Errors in following Categories Found:

Categorie 4 : Inability to make deferense betwen delimiter and pattern when calling “.split“ method for string class.

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

```
require 'csv'

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

```
        name[0] = name[0].to_s

        name[1] = name[1].to_s

        if name.size == 3
                if name[0].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
```

Program is in Rank : 2.Where 1 is worst and 5 is best.

        # It is not easy to understand, read and work with this program.

Appendix for Petko_Bozhinov from A class:

```
# 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|
                csv << pusher
        end
end
```

Problem : Errors in following Categories Found:

No Category Match Errors found in this code

Errors Found:

No Need of function numeric?

```
require 'csv'
class String
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[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

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

```
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Radoslav_Kostadinov form A class:

=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

```
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
```

Problem : Errors in following Categories Found:

No Category Match Errors found in this code

Errors Found: There was need of 1 Hash and 1 Array in order to maintain the file_names
at their output.

```
require 'csv'
file1 = Hash.new
file2 = Hash.new
hash = Hash.new
path1 = ARGV[0]
path2 = ARGV[1]
array = Array.new
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

file1.each_pair do |first_name,last_name|
        file2.each_pair do |first_name1,last_name1|
                if first_name1 == first_name and last_name1 == last_name
                else
                        hash[last_name1] = first_name1
                end
        end
end

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

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Borislav_Rusinov from A class:

```
=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
a=ARGV[0]
require 'csv'
array=[]
Dir.glob("#{a}*.*")  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
```

Problem : Errors in following Categories Found:

```
        # Categorie 1 : Wrong csv.file name.

        # Categorie 6 : Inability to read task correctly.
```

```
a=ARGV[0]
require 'csv'
array=[]
Dir.glob("#{a}*.*")  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("result.csv", "w") do |csv|
        array.each do |arg|
        csv.puts(arg)
        end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Dimitar_Nesterov from A class:

=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

```
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(' ')
	}
}
```

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

Also there was need of 1 Hash map.

```
require 'csv'
arr = Array.new
h = Hash.new
Dir.glob("#{ARGV[0]}*.rb*") do |file|
	file_str = file.split(/\//).last
	if(file_str.split(/_/)[1].size == 5)
		h[file_str.split(/_/)[0].to_s] = file_str.split(/_/)[1].to_s
	end
end
CSV.open('result.csv','w') do |csv|
	arr = h.sort_by{|k,v|v}
	arr.each do |e|
		csv << e
	end
end
```

Program is in Rank : 5.Where 1 is worst and 5 is best.

It is easy to understand, read and work with this program.

Appendix for Stanislav_Valkanov form A class:

```
#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)
a["#{name}"] = last
end
end
CSV.open("result.csv", "w") do |csv|
Hash[a.sort.reverse].each do |element|
csv << element
end
end
```

Problem : Errors in following Categories Found:

Categorie 2 : Sorting Hash.map (No knowledge that the RESULT afther sorting Hash is Array.

Categorie 8 : Lexical and Language (Rubby) errors in Syntax. (Found only in results from A class).

```
require 'csv'
ar = Array.new
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|
ar = a.sort.reverse
ar.each do |element|
csv << element
end
end
```

Program is in Rank : 4Where 1 is worst and 5 is best.

It is medium to easy to understand, read and work with this program.

Appendix for Simeon_Shopkin from A class:

=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 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

=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
```

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

To MANY changes were Need to fix this code! that woucld make it verry difrerent from it is original apperience!

Appendix for Nikola_Marinov from A class:

=begin
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 smalles 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
count += 1
end
end
array = array.sort_by{|el| el|0|}

CSV.open("task.csv",w) do |csv|
array=uniq.each do |element|
csv << element
end
end
```

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

To MANY changes were Need to fix this code! that woucld make it verry difrerent from it is original apperience!

Appendix for Lubomir_Yorkov from A class:

Problem : Errors in following Categories Found:

Categorie 7 : No task found in the code (Found only in results from A class).

We can not fix somethink we do not know what it purpose is !

Appendix for Veslin_dachev from A class:

Problem : Errors in following Categories Found:

Categorie 7 : No task found in the code (Found only in results from A class).

We can not fix somethink we do not know what it purpose is !

Appendix for Tihomir _Lidanski from A class:

```
#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 smalles 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

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

To MANY changes were Need to fix this code! that woucld make it verry difrerent from it is original apperience!

Appendix for David_Georgiev from B class:

```
	#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
	CSV.open("result.csv", "w") do |csv|
		students_names.sort.each do |last, first|
			csv << ["#{first}",  "#{last}"]
		end
	end
```

We decided that this code has NO ERRORS and is WRONG Evalueted!!!

Appendix for Ilian_Germanov form B class :

```
=begin
	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
=end

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
                if flag == 0
                        results[lastname1] = firstname1
                end
        end
    end
end

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

We decided that this code has NO ERRORS and is WRONG Evalueted!!!

Appendix for Valdimir_Yordanov:

```
#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| #missing backslash
 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)
 names[first_name] = second_name
 end
```

```
end

names = names.sort
puts names

require 'csv'
CSV.open("results.csv", "w") do |csv| # File name is wrong.It should be
result.csv as it is given in the task.
 names.to_a.each do |element|
 csv << element
 end
end
```

Problem : Errors in following Categories Found:

Categorie 1 : Wrong csv.file name.

Categorie 6 : Inability to read task correctly.

```
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

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

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```

Appendix for Valentin_Varbanov from B class:

```
=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
 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
 studentcsv.push(name[1], name[2]) #Outside of the loop
end
CSV.open("result.csv", "w") do |csv|
 studentcsv.each do |string|
 csv << string
 end
end
```

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

To MANY changes were Need to fix this code! that woucld make it verry difrerent from it is original apperience!

Appendix for Nikolay_Mihailov from B class:

```
#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 smalles 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'
hash = Hash.new
count = 0
 Dir.glob(ARGV[0] + "/*.rb") do |file|

 first = file.split(/\//).last
 puts first

 #for (i = 0;i < first.length;i+=1)
 size = first.length
 i = 0
 first.each do |element|
 print "element"
 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

 end
 end
 puts count
 end

 Dir.glob(ARGV[1] +"/*.rb") do |secFile|
 sec = secFile.split(/\//).last
 #puts sec

 end

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

Problem : Errors in following Categories Found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

To MANY changes were Need to fix this code! that woucld make it verry difrerent from it is original apperience!

Appendix for Veselina_Kolova from B calss:

```
=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


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 #No need of this line.
 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 # csv << [v, k]
 end
end
```

Problem: Errors in the following Categies found:

Categorie 5 : Even the writer of the code does not know what he is doing (We decided to name this categorie : Unfollowable logic).

Categorie 6 : Inability to read task correctly.

```
require 'csv'
people_a = Array.new
people = Hash.new
Dir.glob("#{ARGV[0]}/*.*").each do |text_file|
                        text_file = text_file.split("/").last
                        if (text_file.split("_")[1].length == 5) then
                                people[text_file.split("_")[0]] = text_file.split("_")[1]
                        end
end

people_a = people.sort.reverse

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

end

Program is in Rank : 5.Where 1 is worst and 5 is best.

It is easy to understand, read and work with this program.

Appendix for Stefan_Iliev from B class:

```
#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} ]



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

```
        end
end
```

Problem : Errors in following Categories Found:

        # Categorie 1 : Wrong csv.file name.

        # Categorie 6 : Inability to read task correctly.

```
#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} ]



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

Program is in Rank : 5.Where 1 is worst and 5 is best.

```
        # It is easy to understand, read and  work with this program.
```