Technological School “Electronic Systems”
associated with Technical University Sofia

“Learn from your mistakes!”

Flowerpot code

Denis Trenchev
Stanislav Gospodinov
OCT 2014

Report

In most cases students didn’t understand tasks correct or write wrong the name of csv file. This can be avoided if students read tasks more carefully. Other common kind of mistakes are wrong checking file format.There is also a lot of syntax errors in the programs, such as wrong variable names and wrong statements.

Most of the programs can be fixed easily but others are impossible to fix. There even a few student that didn’t understand their task. If students have good fixtures it will be easy for them to do the programs.

Appendix

Appendix 1

Task:
Develop a program named FirstName_LastName_ClassNumber_5f1c22.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
Code:

```
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
```

Problems:
File name is wrong;
ARGV[0] + "*.rb" isn’t valid argument for Dir.glob() function. It must be replaced with "#{ARGV[0]}*.rb";

Strings shouldn’t be capitalize;
Must check file name format;
Didn’t sort the hash;
Shouldn’t include extensions.

Solution:
There no quick solution.

Rank:
2.5/5

Appendix 2

Task:
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

Code:

```
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
```

Problems:
The program doesn’t match with the task;

name = file.split ("/")last.split(".") is wrong;

Solution:
	There is no quick solution.

Rank:
1/5

Appendix 3

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

Code:

```
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:

Checking file format and string last length is wrong;

Solution:
Should change positions of conditions;
Rank:
4/5

Appendix 4

Task:
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 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
Code:

```
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:

Solution:

Rank:
2/5

Appendix 5

Task:
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
```

Code:

```
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: The csv file is named results.csv.

Solution: Change the csv file name
Rank: 3/5

Appendix 6

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
```

Code:

```
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:

ARGV[0] + "*.rb" isn't valid argument for Dir.glob() function. It must be repalced with "#{ARGV[0]}*.rb";

` Strings should not be capitalized.

It should be sorted by DESC

Solution:

There no quick solution

Rank: 2/5

Appendix 7

Task:

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
```

Code:

```
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: Must check file name format

ARGV[0] + "*.rb" isn't valid argument for Dir.glob() function. It must be repalced with "#{ARGV[0]}*.rb";

Solution:There no quick solution

Rank: 2/5

Appendix 8

Task:

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

Code:

```
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: Must check for file name format.

Solution: The file name format must be (*.rb) not (*.*)

Rank: 2/5

Appendix 9

Task:

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
```

Code:

```
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: Must check for file name format.

The file name must be FirstName_LastName_number.rb

Solution: The file name format must be (*.rb) not (*.*)

Change the file name.

Rank: 4/5

Appendix 10

Task:

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
```

Code:

```
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
```

Problem: The csv file is named results.csv.
        Strings should not be capitalized.
        It should be sorted by DESC.
Solution: Change the csv file name
          Sort by DESC
Rank: 3/5

Appendix 11

Task:
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
```

Code:

```
require 'csv'
hash = Hash.new
Dir.glob("#{ARGV[0]}*.rb") do |file|
	filename = file.split('/').last.split('.').first;
		if filename.split('_').length == 3
			if filename.split('_')[1].length == 5
				hash[filename.split('_')[0]] = filename.split('_')[1]
			end
		end
end
hash = Hash[hash.sort_by{|k, v| v}]
CSV.open("results.csv", "w") do |csv|
	hash.each do |key, value|
		csv << [key, value].flatten
	end
end
```

Problem: The csv file is called results.csv
Solution: Change the csv file name
Rank: 4.5/5

Appendix 12

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
```

Code:

```
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
```

Problems:
Wrong defining method numeric?

Solution:
Removing numeric? method and checking for its result.
Rank:
4/5

Appendix 13

Task:
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

Code:

```
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
```

Problems:

Should search only *.rb files;

array(count) is wrong syntax;

Solution:

array(count) must be replaced with array[count]

Rank:

2.5/5

Appendix 14

Task:

Develop a program named FirstName_LastName_ClassNumber_d8aa65.rb

1. you are given two arguments for a folders with files;

1.1 If there are other arguments they should be discarded;

2. Find all the files from both folders that are not in the format FirsrName_LastName_digits.rb. If there are duplicates the file must be written only once.

2.1 If two files are of the same 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

Code:

```
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= short.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
```

Problems:
Conditions and sorting are wrong
Solution:
Changing wrong conditions
Rank:
2.5/5

Appendix 15

Task:
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
```

Code:

```
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: The csv file is named results.csv.

Solution: : Change the csv file name

Rank: 2/5

Appendix 16

Task:

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
```

Code:

```
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
```

Problem:

The program is incomprehensible

Solution:

There is no quick solution.

Rank: 1/5

Appendix 17

Task:

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
```

Code:

```
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
```

Problem:

Solution:

Rank: /5

Appendix 18

Task:

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

Code:

```
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
```

Problem: Must check for file name format

Solution: The file name format must be (*.rb) not (*.*)

Rank: 2/5

Appendix 19

Task:

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
```

Code:

```
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|

			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: ARGV[0] + "*.rb" isn't valid argument for Dir.glob() function. It must be repalced with "#{ARGV[0]}*.rb";
Wrong comparing char and integer
Solution: There no quick solution
Rank: 2/5

Appendix 20

Task:
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 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
```

Code:

```
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
```

Problem: ARGV[0] + "*.rb" isn't valid argument for Dir.glob() function. It must be repalced with "#{ARGV[0]}*.rb";:

Must check for file name format

Solution: The file name format must be (*.rb) not (*.*)

Rank: 2/5

Appendix 21

Task:

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

Code:

```
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
```

Problems:

Wrong checking for file format;

Wrong csv name

Solution:

There is no quick solution

Rank:

3.5/5

Appendix 22

Task:

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 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

Code:

```
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
```

Problems:
The program works only for one file;
file_neme is split wron;
Must compare ch_count with 7 not with 9;
Solution:
There is no quick solution
Rank:
1.5/5

Appendix 23

Task:
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
Code:

```
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
```

Problems:
Wrong variables names;
Wrongs csv name and sorting;

Solution:
There is no quick solution

Rank 3/5