Technological School “Electronic Systems” associated with Technical University Sofia

Report:

What are the errors and how could this errors be avoided in the first place?

made by Kalin Marinov and Dimitar Matev (Team “Programmable Logics”)

Date: 27.10.2014

Common Problems:

Problem 1:

Wrong output file (Ex.: results.csv, students.csv).

Problem 2:

Not checking the format correctly

Problem 3:

Writing more than ones same student (Ex.: FirstName_SecondName_1.rb and FirstName_SecondName_2.rb).

Problem 4:

No sorting where it's required or it has wrong sorting.

Problem 5:

Attempts to work with nil as string or number.

Problem 6:

Wrong output file content.

Other Problems: Wrong variable names, False method usage, Too much unnecessary lines of code.

Report for 11A class:

Student: Borislav Rusinov

Code:

```ruby
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
```

Comments:

1) writing to a wrong csv
2) wrong check for name

Quickfix:

1) add after dir.glob next unless my_text_file =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/ for name check
2) change results.csv to result.csv

Rate: 4

Student: Denis Trenchev

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
```

Comments:

1) writing to a wrong csv, undefined local variable or method `arr' lines 31,32,45,46
2) undefined local variable or method `name_1' at lines 45,46

Quickfix:

1) change csv to result.csv
2) change arr on lines 31,32 to arr1 and on lines 45,46 to arr2
3) change name_1 to name at lines 45,46

Rate: 3

Student: Dimitur Nestorov

Code:

```ruby
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
```

Comments:

1) undefined method `-@' at line 41

Quickfix:

1) change line 41 to array = array.sort_by {|el| el[1]}.reverse

Rate: 4

Student: Radoslav Konstadinov

Code:

```
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
```

Comments:

1) more than ones writing the same name

Quickfix:

1) after the file1.sort in csv.open the code should look something like this

```
check = true
file2.sort.each do |first_name1, last_name1|
        if first_name1 == first_name and last_name1 ==
last_name
                check = false
        end
end
if check
        csv << [last_name, first_name]
end
```

Rate: 3

Student: Kristina Pironkova

Code:

```ruby
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
```

Comments:

1) wrong output csv
2) not checking the file form

Quickfix:

1) change csv output to result.csv
2) add after dir.glob next unless file_name =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/ for file form

Rate: 4

Student: Ivelin Slavchev

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
```

Comments:

1) undefined local variable or method `digit' at line 33
2) wrong checking for file form

Quickfix:

1) change digit at line 33 to digits
2) add after every dir.glob next if (file1 or file2) =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/ for file form

Rate: 4

Student: Ivo Vulchev

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
```

Comments:

1) no 'and',! at line 23
2) no space at line 30 between 'and' and !hash_fold2.include?(name[0])

Quickfix:

1) change digit at line 33 to digits
2) change at line 23 'and !hash_fold1.include?(name[0])'
3) add space at line 30 between 'and' and !hash_fold2.include?(name[0])

Rate: 4

Student: Kalin Marinov

Code:

```ruby
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
```

Comments:

1) wrong sort
2) at line 31 'do' is on second place
3) not checking file form
4) wrong content of output file

Quickfix:

1) change digit at line 33 to digits
2) at line 31 change to hash.sort_by {| k, v | k }.each do |key|
3) add after dir.glob next unless name =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/
4) change line 32 to csv << [key[0].split("_").first,key[1]]

Rate: 3

Student: Lubomir Yankov

Code:

```ruby
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
```

Comments:

1) wrong output file
2) wrong content in output file
3) counting 9 digits not 7

Quickfix:

1) change output file to result.csv
2) change at line 22 -> 9 to 7
3) change array.each loop to:

```
array.each do |element|
        temp = ""
        element[0].each do |word|
                temp += "#{ word }"
        end
        csv << [temp,temp.length/2]

end
```

Rate: 2

Student: Petko Bozhinov

Code:

```ruby
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
```

Comments:

1) not checking file form
2) wrong content csv output

Quickfix:

1) add after every dir.glob next unless (file or file2) =~ [0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/

2) change line 46 to output.uniq.each do |pusher|

Rate: 4

Student: Stanislav Valkanov

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
```

Comments:

1) not full checking file form
2) if first name of a student is equal to first name of another student it count them as a one person

Quickfix:

1) add after dir.glob next unless my_text_file =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/
2) use as a key first name + second name

Rate: 4

Student: Stanislav Gospodinov

Code:

```ruby
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
```

Comments:

1) not full checking file form
2) wrong output file
3) if first name of a student is equal to first name of another student it count them as a one person

Quickfix:

1) change output file to result.csv
2) add after dir.glob next unless file =~ /[0-9a-zA-Z]+_[0-9a-zA-Z]+_\d+\.rb\z/
3) use as a key first name + second name

Rate: 4

Student: Simeon Shopkin

Code:

```ruby
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
```

Comments:

undefined method `split' at line 23
not full checking file form
at line 27 first_files != second_files will always be true
trying to split an array at line 29
if one person has more than two files it will save them both

Quickfix: more than half of the code must be fixed

Rate: 2

Report for 11Б class:

Student: Borislav Stratev

Code:

```ruby
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
```

Comments:

1) no checks
2) wrong output file name("results.csv")
3) unnecessary actions
4) incorrect sorting

Quickfix: None

Rate: 2

Student: David Georgiev

Code:

```ruby
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
```

Comments:

1) no checks for correct filename
2) error in .length method

Quickfix:

1) 20:if name[1].length == 5 -> if name[1].to_s.length == 5

Rate: 4

Student: iliyan Germanov

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
```

Comments:

1) not enough checks

Quickfix:

1) 25:if filename1.count("_") == 2 -> if filename1 =~ /\A[0-9A-Za-z]+_[0-9A-Za-z]+_\d+.rb\z/
2) 33:if filename2.count("_") == 2 -> if filename2 =~ /\A[0-9A-Za-z]+_[0-9A-Za-z]+_\d+.rb\z/

Rate: 4

Student: Lili Karakoleva

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
```

Comments:

1) left extensions,no sorting,no duplicates removal
2) some unnecessary actions

Quickfix:

1) last_name(_1).split('.').first

2) 38:student1.each do |fn, ln| -> student.sort.uniq.each do |fn, ln|
3) 40:student.each do |fn1, ln1| -> student1.sort.uniq.each do |fn1, ln1|
4) checker (ch=0) after student.each becoming 1 if theres a file matching it at 40. and writing if it's 0

Rate: 3

Student:Nikolay_Mihailov

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
```

Comments:

1) too C
2) not compiling
3) not finished

Quickfix: None

Rate: 1

Student: Stanislav Iliev

Code:

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

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

Comments:

1) no checks for acceptable file
2) too C

3) checks are made only from one file to one other file. Impossible to determine if theres match without checking against all
4) There is overwriting so part of the data is lost

Quickfix:

1) involves rewriting the whole second half of the code

Rate: 2

Student: Stefan Iliev

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
```

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

Comments:

1) Writing in results.csv instead of result.csv
2) The code is a bit too longxx
3) Dir.glob gose to sub folders as well

Quickfix:

1) 56:CSV.open("results.csv","w") do |csv| -> CSV.open("result.csv","w") do |csv|

Rate: 5

Student: Valentin Varanov

Code:

```ruby
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])
end

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

Comments:

1) separately checking the first and the last name - theres possibility of equal first or last names but not the combination of the two
2) name and name2 are local arrays they must be declared earlier or they'll be out of scope
3) no require 'csv'
4) String passed to csv method << instead of array or hash
5) no last name sort
6) only writing first name to file

Quickfix:

1) require 'csv'
2) name=[] name2=[] before inner loop
3) name check -> "#{name[0]} #{name[1]}" == "#{name2[0]} #{name2[1]}"
4) name sort -> studentscsv.sort_by{|fn,ln| ln}

Rate: 3

Student: Veselina Kolova

Code:

```ruby
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
```

Comments:

1) syntax errors
2) unnecessary if,
3) hash with key last name will remove people with same surnames

Quickfix: None

Rate: 2

Student: Vladimir Yordanov

Code:

```
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
```

Comments:

1) unnecessary dealing with ARGV[1]
2) no format check
3) hash with key last name will remove people with same first names
4) writing in results.csv instead of result.csv
5) many variables

Quickfix: None

Rate: 3