TECHNOLOGICAL SCHOOL
"ELECTRONIC SYSTEMS" –
SOFIA

R E P O R T

TEAM INNOVATION

Atanaska Ivancheva
Valentin Georgiev
Veselina Kolova

October 28th 2014

We're given the task of investigating and learning from the errors classes A and B made during their test held on October 16th 2014 and write a report about it.

There're a lot of errors found in the programs. Sadly, a very little part of the student body had successfully finished and submitted their works. Others were really close to doing so, but came out to not be so lucky as something didn't allow them to do it. We will look more closely at the already mentioned errors in the paragraphs below.

Most of the most common mistakes were made due to not fully reading or fast skipping through the task's body. Such errors are from the type "result-results", meaning the students produced a result from their given problem in CSV files named "results.csv" instead of "result.csv". Of course, there are students that have chosen a completely different file name, something like "task.csv". There are also incorrectly named files. We believe that working more often with different file name examples in class as well at students' homeworks' tasks would have prevented that from happening and made students pay more attention to the problems they are given to solve.

Part of the students simply could not figure out what was asked of them and what did they had to do. It had had caused a lot of confusion. There were some mistakes in the tasks that made them hard to understand and given the fact that a group of people has problem with the English language - it's their third foreign language or they just hadn't studied it properly - also arise uneasiness. Giving more but easier tasks for home or classwork would solve the problem. Involving some technical literature in the English Language classes would also do good for the students.

Mass of the mistakes were made simply because the students didn't know what exactly was happening in the language - meaning lack of technical knowledge. We are not sure if that's a fault of the teacher's body or the students didn't do their homework or simply are not paying attention in class. We guess that, really, it is not the teacher that has to be blamed, since most of the programs submitted, yes, did have a lot of errors, but the said errors were from small magnitude - something made unconsciously while they're typing but their brains were thinking five or more lines ahead. Frequently met error is writing wrong number backslashes or using plus sign when using Dir.glob and handling with Ruby arguments.

Of course some of the errors happen to work on quite a number of different computers, but when tried to run most throw out errors that are simply hard to understand and disturb the students that used them in their programs.

Other mistakes are also made due to the lack of knowledge of the programming language, or because the students were in a hurry, such as the wrong use of the equal to and direct assignment operators. This may not seem like a big mistake but for sure it is. Sometimes students may be able to correct the cause of an error. Also the students don't know the difference between the methods .length and .size and therefore use improper or unspecified string comparison methods.

When given a specific format for file names to work with, students tend to get confused or faked out by extension methods when they try to check for the extension of the file they need to work with. While there certainly are advantages to using extension methods, they can cause problems and a cry for help for those who aren't aware of them or don't

properly understand them. This is especially true when looking at code samples online, or at any other pre-written code.

A typical way to extract data from text using a regular expression is to use the *.match* method, however students either not know about it and decide to use something that has no logic in it at all, or simply ignore it and do some really mindless things in their code.

Some of the students that have submitted their work had given up at some point and it is hard to figure out what were they trying to do in their program at that time. Other students may have had some idea how to finish their work on the program but were not able to due to lack of time. Pretty often warnings that Terminal had flagged can easily become an error that could make them waste a lot of time while trying to track it down.

The secure copy protocol also adds distress as some people don't understand how to work with it or find it hard to use. Maybe homeworks done online with an online Judge, set timer and use of SCP would help to improve speed and help students work to pay attention more to what is asked of them to do.

There are quite of a number of works that have no task mentioned in the beginning - or really, nowhere in the file - and it's hard, actually impossible, to know what problem the student was trying to solve. There's no way to find out if there are errors if none are thrown out to you by the Terminal.

A good idea is while students are getting ready for the exam to prepare some fixtures and files with examples of programs. This is a good practice because in this way students won't lose precious time while the exam is taking place in making empty documents to test their programs on.

Six of the submitted 34 programs are correct. 18 have and 10 don't have a quick way to be fixed.

Appendixes

Borislav Rusinov:

```
=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! # Wrong put "!". It should be array.sort.
array.reverse! # Wrong put "!". It should be array.reverse.
File.open("results.csv", "w") do |csv| # File name is wrong. It should be
"results.csv" as given in the task.
     array.each do |arg|
     csv.puts(arg)
     end
end
```

Does not cover all variants for an invalid file name. Will need more than a quick fix.
Rating: 4.

Denis Trenchev:

```
=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 # Student does not understand the
task. (Checks if last name is 5 characters, not first name.)
               arr1[i] = []
               arr[i][0] = name[0] # Array "arr" does not exist. Probably
mistaken with arr2?
               arr[i][1] = name[1] #      Array "arr" does not exist.
               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  # Student does not understand the
task. (Checks if last name is 5 characters, not first name.)
               arr1[i] = []
               arr[i][0] = name_1[0]  # Array "arr" does not exist.
               arr[i][1] = name_1[1]  # Array "arr" does not exist.
               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|  # Error in the file name. It should
be "student.csv" as given in the task.
    sort.each do |element|
        csv << element
    end
end
```

There is not an quick solution to this program. There are lots of things that need to be changed so it can work.
Rating: 3.

Dimitar Nestorov:

```
#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
```

Unknown problem with array - does not know how to fix.
Rating: 4.

Dimitar Terziev:

```
=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|  # missing backslash
     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(' ')
     }
}
```

Rating: 4.

Georgi Ivanov:

```
=begin Develop a program named
FirstName_LastName_ClassNumber_871529.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"

arr = []
i = 0

Dir.glob(ARGV[0]+"*.rb") do |file|
     name = file.split('/').last.split('.').first.split('_')
     firstname = name[0]
     lastname = name[1]
     exercise = name[2]

     if firstname == '' || lastname == '' || exercise == ''
     elsif name.length == 3

     if lastname.length == 5
          arr[i] = []
          arr[i][0] = name[0]
          arr[i][1] = name[1]
          i+=1
     end
     end
end


daiba = arr.sort_by{|asd| asd[0]}.reverse!
CSV.open("result.csv", "w") do |csv|
     daiba.each do |element|
```

```
          csv << element
     end
end
```

The program is correct and doesn't need any corrections.
Rating: 4.

Hristo Dachev:

```
=begin
Develop a program named
FirstName_LastName_ClassNumber_4a196f.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'

hash = Hash.new

Dir.glob("#{ARGV[0]}*").each do |path|
     first_name =   path.split("/").last.split("_").first
     last_name =    path.split("/").last.split("_",
2).last.split("_").first
     digit =        path.split("/").last.split("_",
2).last.split("_").last.split(".").first
     name = path.split("/").last

     if name.include? "_" then counter = name.count "_" end

     if (counter != 2) || (digit.to_i.to_s != digit)
          l = name.length
          hash[name] = l
```

```
     end
end
Dir.glob("#{ARGV[1]}*").each do |path|
     first_name =   path.split("/").last.split("_").first
     last_name =    path.split("/").last.split("_",
2).last.split("_").first
     digit =        path.split("/").last.split("_",
2).last.split("_").last.split(".").first



     name = path.split("/").last
     if name.include? "_" then counter = name.count "_" end

     if (counter != 2) || (digit.to_i.to_s != digit)
          l = name.length
          hash[name] = l
     end
end

CSV.open("result.csv", "w") do |csv|
     hash.sort_by{ |k, v| v}.each do |name, length|
          csv << ["#{name}","#{length}"]
     end
end
```

The program is correct and doesn't need any corrections.
Rating: 3.

Ivelin Slavchev:

```
=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) # Suggests if (ext1 != "rb") or
(!digit1.match(/\w/)) or (short1.scan("_").count != 2) as
digit1.to_i.to_s != digit1 pretty much does nothing.
          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) # Suggests if (ext2 != "rb") or
(!digit1.match(/\w/)) or (short2.scan("_").count != 2) as
digit1.to_i.to_s != digit1 pretty much does nothing.
          result[short2] = short2.length
     end
end
result.sort_by{|k, v| v} # The result from sorting is not saved. Suggestion:
result = result.sort_by{|k, v| v}
CSV.open("result.csv", "w") do |csv|
     result.each do |p|
          csv << p
     end
end
```

Program works perfectly well with the pretty fast made suggested changes.
Rating: 5.

Ivo Valchev:

```
=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])# if name[0] and
name[1] and name[0].length == 5 and !isNum!=nil and
!hash_fold1.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
```

Program is fixed and works.
Rating: 5.

Kalin Marinov:

```
#==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 # It should be name =
name.split("/").last.split(".").first.split("_")
     short_name = name.split('_')[1] # It should be  first_name =
name[0]  short_name = name [1]

     if short_name.length == 5
          hash[name] = short_name #It should be hash[first_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
```

Program is fixed and works.
Rating: 4.

Kamena Dacheva:

```
=begin
Develop a program named
FirstName_LastName_ClassNumber_0af18f.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

student = Hash.new { |name, programs| name[programs] = []}
directory = ARGV[0]
require "csv"

class String
  def is_number?
    Float(self) != nil rescue false
  end
end

Dir.glob("#{directory}/*.*") do |my_repository|

    name_dir = my_repository.split("/").last

    name = name_dir.split("_").first.capitalize
    sir_name = name_dir.split("_",
2).last.split("_").first.capitalize
    program = name_dir.split("_").last.split(".").first
    ex = name_dir.split("_").last.split(".").last
```

```
     if name_dir.include? "_" then counter = name_dir.count "_"
end
     student["#{name}"] << sir_name if ((counter == 2) &&
(sir_name.length == 5) && (program.is_number?) && (ex ==
"rb"))
end

CSV.open("result.csv", "w") do |csv|
     student.sort_by{|k, v| v}.reverse.each do |f_name, l_name|
          csv << [f_name,l_name].flatten
     end
end
```

The program is correct and doesn't need any corrections.
Rating: 5.

Kristina Pironkova:

```
=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.spli
t("_").first.capitalize
```

```
          if first_name.length == 10

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

end


CSV.open("results.csv", "w") do |csv| # CSV.open("result.csv",
"w") do |csv|
     results.sort.each do |first,last| # results.sort_by{|k,v|
v}.reverse.each do |first,last|

     csv << [last,first]

     end
end
```

Program is fixed and works.
Rating: 5.

Lubomir Yankov:

```
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
```

It's not clear what task was given to the student.
Rating: 4

Marian Belchev:

```
=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 # Student
does not understand the task.
          hash2[firstName2] = lastName2 + "." + number2 # The
"+" "." + number1" and "+ "." + number2" parts are not needed.
     end
end

CSV.open("results.csv", "w") do |csv| # Error in the file name. It should be
"student.csv" as given in the task.
     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('.',"")] # From the
previous error: csv << [value, key]
          end
          if hash1.has_key?(key) &&
!hash1.has_value?(value.split(".").first) &&
!hash1.has_value?(value.split(".").last.to_i) # and if
!hash1.has_key?(key) &&
!hash1.has_value?(value.split(".").first)
               csv << [key,value.gsub('.',"")]
          end
     end
end
```

Having in mind the changes suggested in the comments in the program, the program works.
Rating: 5

Momchil Angelov:

```
=begin
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
=end
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 # Direct assignment
operator used instead of equal  one.
          next
     else # Unnecessarily used "next" and "else". Student does not fully
understand what's happening.
          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 # Direct assignment
operator used instead of equal  one.

          next
     else   # Unnecessarily used "next" and "else". Student does not fully
understand what's happening.

          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
```

Will need more than a quick fix.
Rating: 3.

Moretti Georgiev:

```
=begin
```

```
Develop a program named
FirstName_LastName_ClassNumber_b7f153.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 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'

student = Hash.new

Dir.glob("#{ARGV[0]}*_*_*.rb")     do |file|
     firstName = file.split("/").last.split("_").first
     lastName = file.split("/").last.split("_",
2).last.split("_").first
     digit =
file.split("/").last.split("_").last.split(".").first
     if lastName.length == 10
          student[firstName] = lastName
     end
end

CSV.open("result.csv", "w") do |csv_file|
     student.sort.each do |key, value| #Does not sort correctly,
suggestion: student.sort_by{|f, l| l}.each do |key, value|

          csv_file << ["#{key}, #{value}"]
     end
end
```

Program is fixed and works.
Rating: 4

Nikola Marinov:

```
=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) = [] # Incorrect use of (). Should be: array[count]
 array(count) [0]=full_name # Incorrect use of (). Should be:
array[count][0]
 array(count)[1]= full_name.to_s.lenght # Incorrect use of (). Should be:
```

```
array[count][1]
 count += 1
 end
 end
 array = array.sort_by{|el| el|0|}

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

Besides the found errors, the reason behind why `.split` does not work is a mystery. There's no quick solution.
Rating: 5.

Petko Bozhinov:

```
# 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| # Student does not sort by Last name:
output.each do |p, f|
          csv << pusher #          csv << [f,p]
     end
end
```

With the two corrections in mind, it's easy to make an quick fix. Program works very good. Rating: 5.

Radoslav Kostadinov:

```
=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("_").firs
t


          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("_").firs
t

          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 # Replaced with: file2.delete(first_name1)
                end
                else # Deleted.
                     csv << [last_name1, first_name1] # Deleted.
                end   # Deleted.
          end
     end
=begin
Included:
file2.sort.each do |first_name,last_name|
      csv << [last_name, first_name]
=end
     end
```

Program works fine with the made changes.
Rating: 5.

Simeon Shopkin:

```
=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| # Terminal
throws out an error, fixed with: Dir.glob("#{ARGV[1]}/*.rb") do
|second_files|
               first_files =
first_files.split("/").last.split(".").first.split("_")
               if first_files.size != 3 # Replace first_files with
first_file
                    if first_files != second_files # Replace
first_files with first_file
                              print_count =
first_files.split("/").last.split(".").first
                              p = print_count.size.to_s
                              print =
first_files[0].capitalize+"_"+first_files[1].capitalize+"_"+fi
rst_files[2]+","+p
                              arr.push(print)
                    end
               end
          end
     end

     CSV.open("result.csv","w") do |csv|
          arr.sort.each do |element|     # Doesn't sort as told to in task:
arr.sort_by{|k, v| v}.each do |element|
               csv << [element]
          end
     end
```

Program does not have quick solution.
Rating: 3.

Stanimir Bogdanov:

```
# scp MyFile.txt
student11b@172.16.18.14:/home/student11b/results_a

=begin
Develop a program named
FirstName_LastName_ClassNumber_ca514d.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'

directory = ARGV[0]
students = Hash.new

Dir.glob("#{directory}*") do |filename|
  unless (filename.split('/').last =~ /^[a-zA-Z0-9]+_[a-zA-Z0-
9]+_[0-9]+.rb$/).nil?
    first_name = filename.split('/').last.split('_')[0]
    second_name = filename.split('/').last.split('_')[1]
    students[first_name] = second_name if first_name.length ==
10
  end
end

CSV.open("result.csv", "w") do |csv|
  Hash[students.sort_by { |first, last| last }.reverse].each
do |first, last|
    csv << [ first, last ]
    # puts "#{first},#{last}"
  end
end
```

The program is correct and doesn't need any corrections.
Rating: 5.

Stanislav Gospodinov:

```
=begin
Develop a program named
FirstName_LastName_ClassNumber_b36abb.rb

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

```
1.1 if there are other arguments they should be discarded
2. file names in this folder are in the form
First_Last_digits.rb;
3. find all the students that have 5 letters in their second
name;
4. Sort the result by Last Name ASC.
5. Produce a result in CSV format named result.csv:

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

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| #File name is wrong. It should be
"results.csv" as given in the task.

     hash.each do |key, value|
          csv << [key, value].flatten
     end
end
```

Program is fixed and works.
Rating: 4.

Stanislav Valkanov:

```
#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 # Suggestion: lastl = last.to_s.length [1]
if (last.length == 5)&&(short_name.split("_").size == 3) # if
(lastl == 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
```

Program's logic is easy to understand but program was submitted with little to none formating, therefore hard to read.

Rating: 4.

[1] `.length` method does not always work and that's the reason why the needed changes are made.

Tihomir Lidanski

```
#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
```

The pogram is not finished
Rating: 3.

Veselin Dechev:

```
=begin
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
=end
```

```
require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*.rb").each do |first| #Missing backslash.
     name1 = first.split("/").last.capitalize
     puts name1
     first_name = name1.split("_").first.capitalize
     puts first_name
     last_name =
name1.split("_",2).last.split('_').first.capitalize
     puts last_name
     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
```

Wrong file name.
Rating: 3.

Borislav Stratev:

```
#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 # Does not
understand the task. Shouldn't merge with 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 # Does not understand the task. Shouldn't merge with 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| #Wrong file name. It should be
result.csv.
     a = h.sort
     a.each do |element|
          csv << element
     end
end
```

Rating: 4.

David Georgiev:

```
#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[0].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
```

The program is correct and doesn't need any corrections.
Rating: 5.

Iliyan Germanov:

```
=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
```

The program is correct and doesn't need any corrections.
Rating: 5

Lili Kokalova:

```
=begin
Develop a program named
FirstName_LastName_ClassNumber_e0ea9c.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form
First_Last_digits.rb;
3. find  the students that are only in the second folder and
not in the first. A student is in both folders if it there is
a file with the same First and Last Name. Digits might be
different;
4. Sort the result by First name ;
5. Produce a result in CSV format named result.csv:

     LastName1,FirstName1
     LastName2,FirstName2
     ...
```

```
     LastNameN,FirstNameN
=end

require 'csv'

student = Array.new
student1 = Array.new

Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
     file_name = file_name1.split("/").last
     first_name = file_name.split("/").last.split("_").first
     p first_name
     last_name =
file_name.split("/").last.split("_",2).last.split("_").first
     #task = file_name.split("_").last.split(".").first.
     student << ["#{first_name}", "#{last_name}"]
end

Dir.glob(ARGV[1]+"/**/*.*").each do |file_name1|
     file_name = file_name1.split("/").last
     first_name = file_name.split("/").last.split("_").first
     p first_name
     last_name =
file_name.split("/").last.split("_",2).last.split("_").first
     #task = file_name.split("_").last.split(".").first.
     student1 << ["#{first_name}", "#{last_name}"]
end

CSV.open("result.csv", "w") do |csv|
     student.each do |fn, ln|
          student1.each do |fn1, ln1|
                    if fn != fn1
                    if ln != ln1 #Should require an array to store the
names,so that there will be no duplicates.
                         csv << ["#{fn1}", "#{ln1}"]
                    end
               end
          end
     end
end
```

Rating: 4.

Nikolay Mihailov:

```
#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
```

The program is throwing mistakes for undefined method .each. Should ask if count==7 and divide it by 2. The program is not ready and is unable to be quick fixed.
Rating: 4.

Stanislav Iliev:

```
#Develop a program named
FirstName_LastName_ClassNumber_627d43.r#
#
#1. you are given two arguments for a folders with files;
#1.1 if there are other arguments they should be discarded;
#2. file names in this folders are in the form
First_Last_digits.rb;
#3. find  the students that are only in the first folder and
not in the second. A student is in both folders if it there is
a file with the same First and Last #Name. Digits might be
different;
#4. Sort the result by Last name ;
#5. Produce a result in CSV format named result.csv:
#
#    LastName1,FirstName1
#    LastName2,FirstName2
#    ...
#    LastNameN,FirstNameN



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

Dir.glob("#{dir1}/*.*") do |file|
     name_array[i] = file.split(/\//).last.split(".").first
     # It should be name_array[i] =
file.split(/\//).last.split(".").first.split("_",2).first
     i += 1
end
```

```
count = i
i = 0
Dir.glob("#{dir2}/*.*") do |file2|
     name_array2[i] = file2.split(/\//).last.split(".").first
     #It should be name_array2[i]
=file2.split(/\//).last.split(".").first.split("_",2).first
     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
               CSV.open("result.csv", "w") do |csv|
                    support_array.each do |element|
                         csv << [element]
                    end
               end
               #Wrong printing! Not like given in the task.
               CSV.open("result.csv", "w") do |csv|
                    support_array2.each do |element2|
                              csv << [element2]
                    end
               end
     end
end
```

Rating: 4.

Stefan Iliev:

```
#Develop a program named
FirstName_LastName_ClassNumber_d77aee.rb
#
#1. you are given two arguments for a folders with files;
#1.1 if there are other arguments they should be discarded;
#2. Find all the files from both folders that are not in the
format FirsrName_LastName_digit.rb. If there are duplicates
the file #must be written only once. If two files are of the
same lenght those files should be sorted in ASC order;
#3. Calculate the length of their names (including
extensions).;
#4. Sort the result by length ;
#5. Produce a result in CSV format named result.csv:
#
```

```
#              File1,3
#              File2,4
#              ...
#              FileN,3

require 'csv'

names_hash = Hash.new

Dir.glob(ARGV[0]+"/*.*").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 ARGV[1] != "err"
     Dir.glob(ARGV[1]+"/*.*").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| # File name is wrong. It should be result.csv as it is given in the task.
     names_hash.each do |element|
          csv << element
     end
end
```

Rating: 4.

Valentin Varbanov:

```
=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
# Must require 'csv'.
CSV.open("result.csv", "w") do |csv|
     studentcsv.each do |string|
          csv << string
     end
end
```

The program is still throwing mistakes for undefined variable. Unable to quick fix.
Rating: 3.

Veselina Kolova:

```
=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  #This end is not needed if the upper line is removed
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
```

Fixed.
Rating: 4.

Vladimir 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
```

Rating: 4.