DATE: OCTOBER 2014

D&K

NAME OF TEAM: D&K
DAVID GEORGIEV
KONSTANTIN VEKILSKI

How many errors are there?
47 errors

How many correct results are there?
5 correct and 8 almost correct results are there.

Are there common errors?
Average of 5 in category

Categories of errors

==========
1. Errors in the output of the program and the .csv file (19)
==========
Wrong output file
Unwanted space "# {name [1] .to_s}"
No data in .csv file
No data in .csv file
No data in .csv file
Missing require 'csv'
Incorrect sorting and export in csv
wrong sort

When two first names are identical returns Ex: Input FirstName1_LastName1 FirstName1_LastName2 --> Exit --> FirstName1_LastName1_LastName2

exporting in results.csv
Wrong csv file - exporting in results.csv And actually nothing export
Nothing appears in the .csv file
Nothing is displayed
Write in results.csv instead result.csv
results.csv instead result.csv
Does not display anything in result.csv
Write in results.csv instead result.csv
Missing require 'csv'
results.csv instead result.csv
==========

==========
2. wrong sorting (5)
==========
Not sorted by Last Name
Not sorted result by first name
Calling name = myfile.split('/')last.split(/\./).first instead of name = myfile.split('/').last.split(/\./).first
incorrect output - First_Last_digits.rb Instead LastName1, FirstName1
wrong sorting
==========

==========

3. ERRORS RELATED TO THE TERMS OF TASKS (2)

==========

no subject
no subject

==========

==========

4. unnecessary complication (2)

==========

Wrong sequence of actions and complication with unnecessary actions
Complicating of the code with text_file.include?(".rb") We can simply change Dir.glob("#{ARGV[0]}/**/*.*").each do |text_file| to Dir.glob("#{ARGV[0]}/**/*.rb").each do |text_file|

==========

==========

5. MISUSE OF METHODS (3)

==========

Calling .split array
Using .length instead of .size
.length Use for array instead for a string

==========

==========

6. syntax errors (4)

==========

Wrong split by point - name = second_folder.split('/').last.split('.').first.split('_')
syntax error, unexpected tIDENTIFIER, expecting kTHEN or ':' or '\ n' or ';'
Many syntax errors
Incorrect split by point filename.split('.')[0]

==========

==========

7. illogical actions (2)

==========

Mistake with compare - if name[1].length == 10 Must be if name[0].length == 10
Useless comparison first_files != second_files always returning true

==========

==========

8. MISSED CHECKINGS (3)

==========

Do not check if the file is *.rb
Is omitted if not student.include?
Does not check whether digits are contained in the file name - in the condition if element == 0 || element == 1 || element == 2 || element == 3 || element == 4 || element == 5 || element == 6 || element == 7 || element == 8 || element == 9 never enters in it

==========

==========

9. undefined method (5)

==========

undefined local variable or method `digit' for main:Object (NameError)
undefined local variable or method `short' for main:Object (NameError)
undefined method `length' for nil:NilClass (NoMethodError)
undefined method `compare_by_identity' for {}:Hash (NoMethodError)
undefined local variable or method `name' for main:Object (NameError)

==========

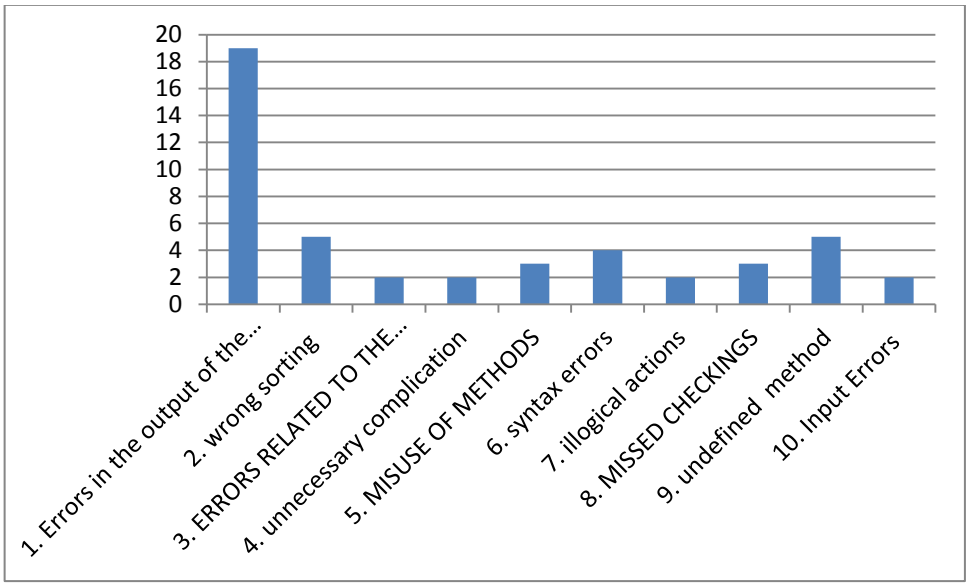
==========

10. Input Errors (2)

==========

Is not reading the correct directory (Missed '/')
Missed dash

==========



How could errors like this be avoided in the first place?
By keeping the syntax of the language and careful reading of the task and try to find the simplest solution for the problem by comparing it with the real world.

11b/

Borislav_Stratev_2_a65be5.rb

```
#Develop a program named FirstName_LastName_ClassNumber_a65be5.rb

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

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

require 'csv'
a = Array.new
h = Hash.new
Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
    Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|

        file_name_1 = dir_file_name_1.split(/\//).last.to_s
        file_name_2 = dir_file_name_2.split(/\//).last.to_s

        if(file_name_1 != file_name_2)
            file_name = file_name_1
            digit = file_name.split(/_/).last.split(/\./).first.to_s
            first_name = file_name.split(/_/).first.to_s
            full_first_name = first_name + digit
            full_first_name = full_first_name.to_s
            tmp = file_name.split("#{first_name}_")
            full_last_name = tmp.last.split(/_/).first.to_s + digit
            full_last_name = full_last_name.to_s
            h[full_last_name] = full_first_name

        end
    end
end

CSV.open("results.csv", "w") do |csv| #must be CSV.open("result.csv"...
    a = h.sort
    a.each do |element|
        csv << element
    end
end
```

results.csv instead result.csv
Can be fixed if we check the file names.
Or rewrite ex:

```
require 'csv'
file1 = []
file2 = []

path1 = ARGV[0]
path2 = ARGV[1]
```

```
Dir.glob("#{path2}/*.rb") do |my_text_file|

 s = my_text_file.split(/\//).last.split("_")
 first_name = s[0]
 last_name = s[1]

 if not file1.include?(["#{first_name}".capitalize , "#{last_name}".capitalize]) then
  file1 << (["#{first_name}".capitalize , "#{last_name}".capitalize])
 end
end
Dir.glob("#{path1}/*.rb") do |my_text_file|

 s = my_text_file.split(/\//).last.split("_")
 first_name = s[0]
 last_name = s[1]

 if not file1.include?(["#{first_name}".capitalize , "#{last_name}".capitalize]) and not file2.include?(["#{first_name}".capitalize ,
"#{last_name}".capitalize]) then
  file2 << (["#{first_name}".capitalize , "#{last_name}".capitalize])
 end

end

CSV.open("result.csv", "w") do |csv|
file2.sort.each do |last, first|
csv << ["#{last}",  "#{first}"]
end
end
```

Easy to understand, not working.
GRADE: 2

David_Georgiev_12_1eea4f.rb

```
    #Develop a program named FirstName_LastName_ClassNumber_1eea4f.rb

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

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

    require 'csv'
    students_names = []
    Dir.glob("#{ARGV[0]}/**/*.rb") do |current_file|

    name = current_file.split('/').last.split(/_/)
# if name[1] == nil
    if name[1].length == 5
        if not students_names.include?(["#{name[1]}", "#{name[0]}"]) then
            students_names << (["#{name[1]}", "#{name[0]}"])
        end
    end
    end
# end
    CSV.open("result.csv", "w") do |csv|
        students_names.sort.each do |last, first|
            csv << ["#{first}",  "#{last}"]
        end
```

```
    end
```

Does not compile.
undefined method `length' for nil:NilClass (NoMethodError)

Can be fixed if we check the file names.

```
if name[1] != nil
  if name[1].length == 5
   if not students_names.include?(["#{name[1]}", "#{name[0]}"]) then
    students_names << (["#{name[1]}", "#{name[0]}"])
   end
  end
 end
```

Easy to understand.
GRADE: 4

Iliyan_Germanov_17_f8b0d9.rb

```
=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 #remove it
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
```

Does not compile.
undefined method `compare_by_identity' for {}:Hash (NoMethodError)
Can be fixed if we remove *results.compare_by_identity* .
Not that easy to understand.
GRADE: 4

```
=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}"] #if not student.include?(["#{first_name}".capitalize, "#{last_name}".capitalize]) student << (["#{first_name}".capitalize, "#{last_name}".capitalize]) end
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}"] #if not student1.include?(["#{first_name}".capitalize, "#{last_name}".capitalize]) student1 << (["#{first_name}".capitalize, "#{last_name}".capitalize]) end
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

CSV.open("result.csv", "w") do |csv|
 student1.sort.each do |first, last|
```

```
 csv << ["#{last}",  "#{first}"]
 end
end
```

Can be fixed if we check the file names.
Not that easy to understand, not working.
GRADE: 2

```
#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|

            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 will never change
            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
```

result.csv empty
Not that easy to understand.
GRADE: 2

```
#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
    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
```

Wrong output *First_Last_digits.rb* expected
LastName1,FirstName1
Easy to understand.
GRADE: 4

Stefan_Iliev_28_d77aee.rb

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

```
#5. Produce a result in CSV format named result.csv:
#
#               File1,3
#               File2,4
#               ...
#               FileN,3

require 'csv'

first_folder = ARGV.shift
second_folder = ARGV.shift || "err"
names_hash = Hash.new

Dir.glob(first_folder+"/*.*").each do |text_file|
    text_file = text_file.split("/").last
    if (text_file.split("_").length == 3) then
        first_name = text_file.split("_")[0]
        second_name = text_file.split("_")[1]
        diggit = text_file.split("_")[2].split(/\./).first
        if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
        if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
        if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
    else
        names_hash[text_file] = text_file.length
    end
end

if second_folder != "err"
    Dir.glob(second_folder+"/*.*").each do |text_file|
        text_file = text_file.split("/").last
        if (text_file.split("_").length == 3) then
            first_name = text_file.split("_")[0]
            second_name = text_file.split("_")[1]
            diggit = text_file.split("_")[2].split(/\./).first
            if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
            if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
            if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
        else
            names_hash[text_file] = text_file.length
        end
    end
end

names_hash = Hash[names_hash.sort_by{|k,v| k} ]
names_hash = Hash[names_hash.sort_by{|k,v| v} ]

puts names_hash

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

results.csv Instead result.csv
Easy to understand.
GRADE: 5

Valentin_Varbanov_4_041472.rb

```
=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])
end

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

Does not compile.
undefined local variable or method `name' for main:Object (NameError)
Missed require 'csv'
We can fix it by
defining the var. name globally
Fix the record in the array and output the results to csv file
Not that easy to understand.
GRADE: 1

Veselina_Kolova_8_65630e.rb

```
=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
        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| #must be CSV.open("result.csv"...
  people.each do |element|
  csv << element
  end
end
```

Not sorting right.
Complicating of the code with text_file.include?(".rb") We can simply change Dir.glob("#{ARGV[0]}//*.*").each do |text_file| to Dir.glob("#{ARGV[0]}/**/*.rb").each do |text_file|**

We can remove
if File.extname(text_file) text_file.include?(".rb") && text_file.split(/_/).last.split(/\./).first.to_i.is_a Integer then
In which the program is running, but still not sorted correctly

Easy to understand.
GRADE: 4

Vladimir_Yordanov_9_4bbed0.rb

```
#Develop a program named FirstName_LastName_ClassNumber_4bbed0.rb

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

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

```
names = Hash.new
Dir.glob (ARGV[0] + "*.rb") do |file|
    if (ARGV[1] == true)
        ARGV[1] == false
    end

    slice = file.split("/").last
    first_name = slice.split('_')[0]
    second_name = slice.split('_')[1]
    if (second_name.length == 5)
        #print first_name
        #puts second_name
        names[first_name] = second_name
    end

end

names = names.sort
puts names

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

Not sorting right.
results.csv instead result.csv
We can fix it by
Changing
Dir.glob (ARGV[0] + "/*.rb") do |file|
to
Dir.glob (ARGV[0]+"/*.rb") do |file|
And by fixing sorting
Easy to understand.
GRADE: 3

Lili_Kokalova_22_e0ea9c.rb

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

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

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

require 'csv'
student = Array.new
student1 = Array.new

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

```
end

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

CSV.open("result.csv", "w") do |csv|
    student.each do |fn, ln|
        student1.each do |fn1, ln1|
            if fn != fn1
                if ln != ln1
                    csv << ["#{fn1}", "#{ln1}"]
                end
            end
        end
    end
end
```

omitted if not student.include?

We can fix it by
In the first Dir.glob
Instead of

```
student1 << ["#{first_name}", "#{last_name}"]
```

write

```
if not student.include?(["#{first_name}".capitalize, "#{last_name}".capitalize])
 student << (["#{first_name}".capitalize, "#{last_name}".capitalize])
end
```

In the second Dir.glob
Instead of

```
student1 << ["#{first_name}", "#{last_name}"]
```

write

```
if not student.include?(["#{first_name}".capitalize, "#{last_name}".capitalize])
 student1 << (["#{first_name}".capitalize, "#{last_name}".capitalize])
```

And fix the output

```
CSV.open("result.csv", "w") do |csv|
 student1.sort.each do |first, last|
  csv << ["#{last}",  "#{first}"]
 end
end
```

GRADE 2/5

Nikolay_Mihailov_25_f70059.rb

#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|

            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
```

Nothing appears in the .csv file
Does not check whether digits are contained in the file name - in the condition if element == 0 || element == 1 || element == 2 || element == 3 || element == 4 || element == 5 || element == 6 || element == 7 || element == 8 || element == 9 never enters
We can fix it by using if element =~ /\d/
Grade 2/5

Stanislav_Iliev_26_627d43.rb

```
#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
    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
```

Wrong output
First_Last_digits.rb
Instead of
LastName1,FirstName1
Grade 4/5

11a/

Borislav_Rusinov_2_6fb3ad.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_6fb3ad.rb

1. you are given an argument for a folder with files;
1.1 if there are other arguments they should be discarded
2. file names in this folder are in the form First_Last_digits.rb;
3. find all the students that have 10 letters in their first name;
4. Sort the result by Last Name DESC.
5. Produce a result in CSV format named result.csv:

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


=end
a=ARGV[0]
require 'csv'
array=[]
Dir.glob("#{a}*.*")  do |my_text_file|
    name = my_text_file.split("/").last.split(".").first.split("_")
    if name[1]!=nil && name[0].length==10
        array << name[0] + "," + name[1]
    end
end
array.sort!
array.reverse!
File.open("results.csv", "w") do |csv| #must be CSV.open("result.csv"...
    array.each do |arg|
    csv.puts(arg)
    end
end
```

Error in the source file
Can not read the correct directory (no '/')
Not sorted by Last Name

We can fix it by
Change

```
Dir.glob("#{a}*.*")  do |my_text_file|
```

to

```
Dir.glob("#{a}/*.*")  do |my_text_file|
```

Change

```
array.sort!
array.reverse!
File.open("results.csv", "w") do |csv|
 array.each do |arg|
 csv.puts(arg)
 end
end
```

to

```
File.open("result.csv", "w") do |csv|
 array.sort.reverse.each do |last, first|
   csv << ["#{first}",  "#{last}","\n"]
 end
end
```

Not easy to understand.
GRADE: 2

Denis_Trenchev_4_b4c3f5.rb

```
=begin
```

Develop a program named FirstName_LastName_ClassNumber_b4c3f5.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb;
3. find the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
4. Sort the result by Last name ;
5. Produce a result in CSV format named result.csv:

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

```
=end

require 'csv'

i = 0
arr1 = []
arr2 = []
arr3 = []

Dir.glob(ARGV[0]+"*.rb") do |first_folder|
    name = first_folder.split('/').last.split('.').first.split('_')

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

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

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

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


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

Wrong output file
Wrong split by point
name = second_folder.split('/').last.split('.').first.split('_')
Easy to understand, not working.
GRADE: 3

Dimitar_Nestorov_7_0d5526.rb

```
#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 # if name[0].length == 10

            array[count] = []
            array[count][0] = name[0].to_s
            array[count][1] = " #{name[1].to_s}"
            count += 1

        end
    end
end
array = array.sort_by {|el| -el[1]}
CSV.open("result.csv", "w") do |csv|

        array.uniq.each do |e|

            csv << e

        end

end
```

Missed dash
Unwanted space " #{name[1].to_s}"
Mistake with compare - if name[1].length == 10

We can fix it by
Adding dash
Dir.glob(ARGV[0] + "/*.rb") do |file|

Unwanted space `" #{name[1].to_s}"`
It must be `"#{name[1].to_s}"`

Compare `- if name[1].length == 10`
Must be `if name[0].length == 10`

Change
```
array[count][0] = name[0].to_s
 array[count][1] = "#{name[1].to_s}"
```
To
```
 array[count][1] = name[0].to_s
 Array[count][0] = "#{name[1].to_s}"
```

Change
```
array = array.sort_by {|el| -el[1]}
CSV.open("result.csv", "w") do |csv|

 array.uniq.each do |e|

  csv << e

 end

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

Now the script is working

Easy to understand, not working.
GRADE: 3

Dimitar_Terziev_6_88db52.rb

```
=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| #Must be Dir.glob("#{ARGV[0]}/*.rb*"){|file|
    file_str = file.split('/').last
    if(file_str=~/\A[a-zA-Z]+\_[a-zA-Z]+\_\d+\.rb\z/ && file_str.split('_')[1].size == 5)
        arr.push("#{file_str.split('_')[1]} #{file_str.split('_').first}")
    end
}
CSV.open('result.csv','w'){|csv|
    arr.uniq.sort.each{|el|
        csv << "#{el.split(' ').last} #{el.split(' ').first}".split(' ')
    }
}
```

result.csv empty

We can fix it by
Adding dash
Dir.glob("#{ARGV[0]}*.rb*"){|file|
Must be Dir.glob("#{ARGV[0]}/*.rb*"){|file|
Not that easy to understand.
GRADE: 4

Georgi_Ivanov_3_871529.rb

```
=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| #Must be Dir.glob("#{ARGV[0]}/*.rb*"){|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
```

result.csv empty
We can fix it by
Adding dash
Dir.glob("#{ARGV[0]}*.rb*"){|file|
Must be Dir.glob("#{ARGV[0]}/*.rb*"){|file|
GRADE: 5

Hristo_Dachev_29_4a196f.rb

```
=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| #must be Dir.glob("#{ARGV[0]}/*…

    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| #must be Dir.glob("#{ARGV[1]}/*…
    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
```

result.csv empty
We can fix it by
Change
Dir.glob("#{ARGV[0]}*").each do |path|
To
Dir.glob("#{ARGV[0]}/*.*").each do |path|

And
Dir.glob("#{ARGV[1]}*").each do |path|
To
Dir.glob("#{ARGV[1]}/*.*").each do |path|

Now the script is working

Easy to understand.
GRADE: 5

Ivelin_Slavchev_10_835552.rb

```
=begin
    Develop a program named FirstName_LastName_ClassNumber_835552.rb
```

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

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

```
=end

require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*").each do |file1|
    short1 = file1.split("/").last
    ext1 = short1.split(".").last
    names1 = short1.split(".").first
    digit1 = file1.split("_").last
    if (ext1 != "rb") or (digit1.to_i.to_s != digit1) or (short1.scan("_").count != 2)
        result[short1] = short1.length
    end
end
Dir.glob(ARGV[1] + "*").each do |file2|
    short2 = file2.split("/").last
    ext2 = short2.split(".").last
    names2 = short2.split(".").first
    digit2 = file2.split("_").last
    if (ext2 != "rb") or (digit2.to_i.to_s != digit) or (short2.scan("_").count != 2) # if(ext2 != "rb") or (digit2.to_i.to_s != digit1) 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
```

Does not compile.
undefined local variable or method `digit' for main:Object (NameError)
We can fix it by
Changing
if (ext2 != "rb") or (digit2.to_i.to_s != digit) or (short2.scan("_").count != 2)
to
if (ext2 != "rb") or (digit2.to_i.to_s != digit1) or (short2.scan("_").count != 2)

Defining didit1 globaly

Yet returns wrong result
Have to rewrite
Easy to understand.
GRADE: 3

Ivo_Valchev_11_6c8bd9.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_6c8bd9.rb

1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form First_Last_digits.rb;
3. find  the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and
Last Name. Digits might be different;
```

4. Sort the result by Last name ;
5. Produce a result in CSV format named result.csv:

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

hash_fold1={}
hash_fold2={}

Dir.glob("#{ARGV[0]}*.*")  do |file|
        name = file.split("/").last.split(".").first.split("_")
        isNum = Integer(name[2]) rescue nil
        if name[0] and name[1] and name[0].length == 5 and !isNum!=nil hash_fold1.include?(name[0])
            hash_fold1["#{name[1]}"] = "#{name[0]}"
        end
end
Dir.glob("#{ARGV[1]}*.*") do |file|
        name = file.split("/").last.split(".").first.split("_")
        isNum = Integer(name[2]) rescue nil
        if name[0] and name[1] and name[0].length == 5 and !isNum!=nil and!hash_fold2.include?(name[0])
            hash_fold2["#{name[1]}"] = "#{name[0]}"
        end
end
File.open("result.csv", "w") do |csv|
    hash_fold1.sort.map do |key, value|
        if (hash_fold1[key]==hash_fold2[key])
            csv.puts("#{key},#{value}")
        end
    end
end
```

Does not compile.
Missing require 'csv'
syntax error, unexpected tIDENTIFIER, expecting kTHEN or ':' or '\n' or ';'

We can fix it by Adding dash
Dir.glob("#{ARGV[0]}/*.*") do |file|
And here
Dir.glob("#{ARGV[1]}/*.*") do |file|
Adding and
if name[0] and name[1] and name[0].length == 5 and !isNum!=nil and hash_fold1.include?(name[0])

But still no result
Have to rewrite
Not easy to understand.
Must be rewritten.
GRADE: 2

Kalin_Marinov_14_12_bce70c.rb

```
#==begin
#Develop a program named FirstName_LastName_ClassNumber_bce70c.rb
#
#1. you are given an argument for a folder with files;
#1.1 if there are other arguments they should be discarded
#2. file names in this folder are in the form First_Last_digits.rb;
#3. find all the students that have 5 letters in their second name;
#4. Sort the result by First name DESC.
#5. Produce a result in CSV format named result.csv:
#
#       FirstName1,LastName1
#       FirstName2,LastName2
#       ...
#       FirstNameN,LastNameN
#==end
```

```
require 'csv'

hash = Hash.new

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

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

Incorrect sorting and recording in csv
We can fix it by
Replacement of line 25
hash[name] = short_name
with
hash[name.split('_')[0]] = short_name

this nonsense

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

Have to be

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

Now is Working

Not easy to understand.
GRADE: 3

Kamena_Dacheva_13_0af18f.rb

```
=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|
#     student.sort_by{|k, v| k}.reverse.each do |f_name, l_name|
            csv << [f_name,l_name].flatten
      end
end
```

Not sorting right.
We can fix it by

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

but When two first names are identical returns Ex: Input FirstName1_LastName1 FirstName1_LastName2 --> Exit --> FirstName1_LastName1_LastName2

Still not working fine

Easy to understand.
GRADE: 4

Kristina_Pironkova_15_890ba0.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_890ba0.rb

1. you are given an argument for a folder with files;
1.1 if there are other arguments they should be discarded
2. file names in this folder are in the form First_Last_digits.rb;
3. find all the students that have 10 letters in their first name;
4. Sort the result by Last Name DESC.
5. Produce a result in CSV format named result.csv:

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


require 'csv'
results=Hash.new
Directory = ARGV[0]
Dir.glob("#{Directory}/*.rb") do |file_name|

      first_name = file_name.split("/").last.split("_").first.capitalize
      last_name=file_name.split("/").last.split("_",2).last.split("_").first.capitalize
```

```
        if first_name.length == 10

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

end


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

    csv << [last,first]

    end
end
```

Write to results.csv
We can fix it by
Changing
CSV.open("results.csv", "w") do |csv|
to
CSV.open("result.csv", "w") do |csv|
Easy to understand.
GRADE: 5

Lubomir_Yankov_16_650c0b.rb

```
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| #must be CSV.open("result.csv"...

    array.each do |element|

        csv << element

    end

end
```

GRADE: 1

Marian_Belchev_17_ad26e0.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_ad26e0.rb

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

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

require 'csv'

hash1 = Hash.new
hash2 = Hash.new

Dir.glob("#{ARGV[0]}*_*_*.rb") do |file1|
    Dir.glob("#{ARGV[1]}*_*_*.rb") do |file2|
        firstName1      = file1.split("/").last.split("_").first
        lastName1  = file1.split("/").last.split("_", 2).last.split("_").first
        number1 = file1.split("_").last.split(".").first

        firstName2      = file2.split("/").last.split("_").first
        lastName2  = file2.split("/").last.split("_", 2).last.split("_").first
        number2 = file2.split("_").last.split(".").first

        hash1[firstName1] = lastName1 + "." + number1
        hash2[firstName2] = lastName2 + "." + number2
    end
end

CSV.open("results.csv", "w") do |csv| #must be CSV.open("result.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
```

***.csv empty and this csv is results.csv**
We can fix it by
Changing
CSV.open("results.csv", "w") do |csv|
to
CSV.open("result.csv", "w") do |csv|

Changing
Dir.glob("#{ARGV[0]}*_*_*.rb") do |file1|
Dir.glob("#{ARGV[1]}*_*_*.rb") do |file2|
to
Dir.glob("#{ARGV[0]}/*.rb") do |file1|
Dir.glob("#{ARGV[1]}/*.rb") do |file2|

removing number1 and number2

Fix the sorting
csv << [key,value.gsub('.',"")]
to
csv << [value,key.gsub('.',"")]

Now is working

Not that easy to understand.
GRADE: 3

Momchil_Angelov_18_d8aa65.rb

```
=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
        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
```

Does not compile.
undefined local variable or method `short' for main:Object (NameError)
short must be defined globally
Yet nothing appears

Have to rewrite

Not easy to understand.
GRADE: 1

Moretti_Georgiev_19_b7f153.rb

```
=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| # 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|
        csv_file << ["#{key}, #{value}"] #csv_file << ["#{key}","#{vlue}"]

    end
end
```

Nothing appears to .csv file
We can fix it by
Changing

Dir.glob("#{ARGV[0]}*_*_*.rb") do |file|
to
Dir.glob("#{ARGV[0]}/*.rb") do |file|

Changing
student[firstName] = lastName
to
student[lastName] = firstName
To resolve the sortation

Changing
csv_file << ["#{key}, #{value}"]
to
csv_file << ["#{value}","#{key}"]
To not displayed quotes
Now is working
Easy to understand.
GRADE: 3

Nikola_Marinov_20_add57e.rb

```
=begin
1. you are given two arguments for a folders with files;
1.1 if there are other arguments they should be discarded;
2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names excluding extension. If there are duplicates the file must be written only once.;
3. Calculate the length of their names (including extensions) divided by 2 rounded to the smalles number;
4. Sort the result by File name ;
5. Produce a result in CSV format named result.csv:

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

requre 'csv'
def is_numeric(o)
 true if Integer(o) rescue false
 end

 array=[]

count=0
Dir.glob(ARGV[0] + "/**/*.*").each do |file|

 full_name=file.split("/").last
 name = file.split("/").last.split(".").first_split("_")

 if name.lenght != 3 && !is_numeric(name[2])
 array(count) = []
 array(count) [0]=full_name
 array(count)[1]= full_name.to_s.lenght
 count += 1

 end
 end

 Dir.glob(ARGV[0] + "/**/*.*").each do |file|

 full_name=file.split("/").last
 name = file.split("/").last.split(".").first_split("_")

 if name.lenght != 3 && !is_numeric(name[2])
 array(count) = []
 array(count) [0]=full_name
 array(count)[1]= full_name.to_s.lenght
 count += 1
 end
```

```
end
array = array.sort_by{|el| el|0|}

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

Nothing is displayed
Many syntax errors
Have to rewrite
Not easy to understand.
GRADE: 2

Petko_Bozhinov_21_954dc6.rb

```
# Develop a program named FirstName_LastName_ClassNumber_954dc6.rb

# 1. you are given two arguments for a folders with files;
# 1.1 if there are other arguments they should be discarded;
# 2. file names in this folders are in the form First_Last_digits.rb;
# 3. find  the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
# 4. Sort the result by Last name ;
# 5. Produce a result in CSV format named result.csv:

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

require 'csv'

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

output = Array.new
i = 0
Dir.glob(ARGV[0] + "/*") do |file|
    file = file.split('/').last.split('.').first.split('_')
    Dir.glob(ARGV[1] + "/*") do |file2|
        file2 = file2.split('/').last.split('.').first.split('_')
        if "#{file[0]} #{file[1]}" == "#{file2[0]} #{file2[1]}"
            if file[2].numeric?
                if file[0].to_s.length == 5
                    output[i] = Array.new
                    output[i][0] = file[0]
                    output[i][1] = file[1]
                    i+=1
                end
            end
        end
    end
end

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

Not easy to understand.
GRADE: 2

Radoslav_Kostadinov_22_772118.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_772118.rb

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

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

=end

require 'csv'
file1 = Hash.new
file2 = Hash.new

path1 = ARGV[0]
path2 = ARGV[1]

Dir.glob("#{path1}*.rb") do |my_text_file|
        s = my_text_file.split(/\//).last.capitalize
        first_name = my_text_file.split("/").last.split("_").first
        last_name  = my_text_file.split("/").last.split("_",2).last.split("_").first


        if s.count('_') == 2 and !((first_name == "" || first_name == " ") || (last_name == "" || last_name == " "))
                file1[first_name] = last_name
            end
end

Dir.glob("#{path2}*.rb") do |my_text_file|
        s = my_text_file.split(/\//).last.capitalize
        first_name = my_text_file.split("/").last.split("_").first
        last_name  = my_text_file.split("/").last.split("_",2).last.split("_").first

        if s.count('_') == 2 and !((first_name == "" || first_name == " ") || (last_name == "" || last_name == " "))
                file2[first_name] = last_name
            end
end

CSV.open("result.csv", "w") do |csv|
    file1.sort.each do |first_name, last_name|
        file2.sort.each do |first_name1, last_name1|
         if first_name1 == first_name and last_name1 == last_name
            begin
            end
            else
                csv << [last_name1, first_name1]
            end
        end
    end
    end
```

Must be rewritten.
Wrong sequence of actions and complication with unnecessary actions
Not easy to understand.
GRADE: 1

Simeon_Shopkin_23_56a835.rb

```
=begin
Develop a program named FirstName_LastName_ClassNumber_56a835.rb

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

        File1,3
        File2,4
        ...
        FileN,3

=end

require 'csv'

arr = Array.new
    Dir.glob(ARGV[0]+"/*.rb") do |first_files|
        Dir.glob(ARGV[1]+"/*.rb") do |second_files|
            first_files = first_files.split("/").last.split(".").first.split("_")
            if first_files.size != 3
                if first_files != second_files
                        print_count = first_files.split("/").last.split(".").first
                        p = print_count.size.to_s
                        print = first_files[0].capitalize+"_"+first_files[1].capitalize+"_"+first_files[2]+","+p
                        arr.push(print)
                end
            end
        end
    end

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

Must be rewritten.
Calling .split for an array
useless comparison first_files! = second_files always return true
Using .length instead of .size
Have to rewrite it
Not easy to understand.
GRADE: 1

Stanimir_Bogdanov_24_ca514d.rb

```
# 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
```

Do not check if the file is * .rb
We can fix it by
Changing
Dir.glob("#{directory}*") do |filename|
to
Dir.glob("#{directory}//*.rb") do |filename|**
Easy to understand.
GRADE: 4

Stanislav_Gospodinov_26_b36abb.rb

```
=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|
    hash.each do |key, value|
```

```
        csv << [key, value].flatten
    end
end
```

Not sorting right.
Write to results.csv instead result.csv
.length Use for an array instead of a string
Incorrect split in point
filename.split ('.') [0]
We can fix it by
Replacing
filename.split('.')[0]
with
filename.split(/\./)[0]

replacing
Dir.glob("#{ARGV[0]}*.rb") do |file|
with
Dir.glob("#{ARGV[0]}//*.rb") do |file|**
Replacing
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
```
with
```
CSV.open("result.csv", "w") do |csv|
  Hash[hash.sort_by { |first, last| last }.reverse].each do |first, last|
    csv << [ first, last ]

  end
end
```
Not that easy to understand.
GRADE: 3

Stanislav_Valkanov_25_4482c1.rb

```
#Develop a program named FirstName_LastName_ClassNumber_4482c1.rb

#1. you are given an argument for a folder with files;
#1.1 if there are other arguments they should be discarded
#2. file names in this folder are in the form First_Last_digits.rb;
#3. find all the students that have 5 letters in their second name;
#4. Sort the result by First name DESC.
#5. Produce a result in CSV format named result.csv:

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

require 'csv'
a = Hash.new
path = ARGV[0]
Dir.glob(path + "**/*.rb") do |my_text_file|
short_name = my_text_file.split('/').last.split('.').first
name = short_name.split("_")[0]
last = short_name.split("_")[1]
last.to_s
if (last.length == 5)&&(short_name.split("_").size == 3)
a["#{name}"] = last
end
end
CSV.open("result.csv", "w") do |csv|
Hash[a.sort.reverse].each do |element|
```

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

Not sorted result by first name
We can fix it by
Changing

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

to

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

Not easy to understand.
GRADE: 4

Tihomir_Lidanski_27_dafd44.rb

```
#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
```

Calling

```
name = myfile.split('/')last.split(/\./).first
```

Instead of

```
name = myfile.split('/').last.split(/\./).first
```

Must be rewritten.
Not easy to understand.

GRADE: 1

Veselin_Dechev_11A2_5f1c22.rb

```
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
```

GRADE: 1