



# Доклад за резултатите от контролна работа

Технологично Училище „Електронни Системи“

Октоври, 2014

**JustCrew**

Nikola Marinov, Petko Bozhinov, Tihomir Lidanski

## Common mistakes:

- The most common mistake is having the output file wrong named.
- Another mistake that we've noticed is not checking the input filenames enough to make sure that the formats are fulfilling the tasks.
- Another frequent mistake is the wrong sorting of the output data
- In not a few programs there are mistakes caused by the usage of large and complicated one-line statements
- There are a bunch of programs that do not compile due to a syntax error

## Common solutions:

- In 1/2 of the programs the errors are caused by misreading, or misunderstanding. So the most common solution is paying enough attention
- Even though, there are programs that require more time to be corrected. A very frequent solution is checking the splits. If all the splits are working, the chance of mistake is reduced by a half.
- The majority of the non-working programs were going to work if the required amount of checks were performed.

## Facts:

- 34 is the count of the submitted tests
- 24 of them are in class A (respectively 10 are in B)
- 50% of class A and 40% of class B are graded with 3 from 5 points
- 5/5 is the third most frequent result in class A
- 20% of the submitted results from both classes are graded with 4/5

# Appendixes:

/in-depth review of the tasks and the mistakes/

## Class A

### 1. Borislav Rusinov:

#### ○ Borislav's task:

Develop a program named `FirstName_LastName_ClassNumber_6fb3ad.rb`

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

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

#### ○ Performance

Borislav's task result is close to expected. He has done great job, despite the few mistakes I noticed. Program run via `ruby Borislav_Rusinov_2_6fb3ad.rb 6fb3ad/folder`

In my opinion his code was written with having in mind future contributors, so I would value his work with 4 out of 5 points.

## ○ Recommendations

The only mistake I noticed into Borislav's program is that he is not performing the required amount of checks to verify the format of the inputted filenames, so First\_Last\_3.txt is allowed (it doesn't have to be because of the .txt extension). My recommendation is modifying Dir.glob arguments in a way they check only .rb files or performing the check into the code with assigning a variable that keeps the extension.

## ○ Code

```
=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}*.rb") do |my_text_file|
  name = my_text_file.split("/").last.split(".").first.split("_")
  if name[1]!=nil && name[0].length==10
    array << name[0] + "," + name[1]
  end
end
array.sort!
array.reverse!
File.open("results.csv", "w") do |csv|
  array.each do |arg|
    csv.puts(arg)
  end
end
```

## 2. Denis Trenchev:

### ○ Denis' task:

Develop a program named `FirstName_LastName_ClassNumber_b4c3f5.rb`

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

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

### ○ Performance

When running Denis' program via `ruby Denis_Trenchev_4_b4c3f5.rb`

`b4c3f5/folder/` it could not compile correctly, because of error

```
Denis_Trenchev_4_b4c3f5.rb:31:in `block in <main>': undefined local variable
or method `arr' for main:Object (NameError)
```

```
from Denis_Trenchev_4_b4c3f5.rb:25:in `glob'from
```

```
Denis_Trenchev_4_b4c3f5.rb:25:in `<main>'.
```

This error is usually caused by using undefined variable.

Another mistake I've found is being outputted from the compiler as follows:

```
Denis_Trenchev_4_b4c3f5.rb:45:in `block in <main>': undefined local variable
or method `name_1' for main:Object (NameError)
```

```
from Denis_Trenchev_4_b4c3f5.rb:39:in `glob'
```

```
from Denis_Trenchev_4_b4c3f5.rb:39:in `<main>'
```

This error appears when attempt to perform an operation with undefined variable was detected.

And at last, Denis failed in the fifth point from the assignment, which was about naming the output .csv file correctly.

The mistakes Denis have assumed in his program are not critical, but are successfully intercepting the normal performance of the program and although his code is well structured, for the work of the program he deserves no more than 1 point.

## ○ Recommendations

The first error (in `block in <main>': undefined local variable or method `arr' for main:Object (NameError)) could be easily avoided by checking the variables used in the loop. One could easily notice that the code contains the undefined variable **arr**. When changing it to **arr1**, this part of code starts working.

The next error (in `block in <main>': undefined local variable or method `name\_1' for main:Object (NameError)) was located in line 45.

This error is also caused by using an undefined variable. When going through the code, I noticed that the second ARGV-containing loop is copied from the first one with minimal changes, that are not enough to make the program works. My recommendation is when having such task, not to copy loops literally. One should pay attention to the loop purpose and design the code in way it satisfies the task.

The last mistake (.csv filename) is caused by not reading the task carefully.

## ○ Code

*=begin*

*Develop a program named FirstName\_LastName\_ClassNumber\_b4c3f5.rb*

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

*1.1 if there are other arguments they should be discarded;*

*2. file names in this folders are in the form First\_Last\_digits.rb;*

*3. find the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;*

*4. Sort the result by Last name ;*

*5. Produce a result in CSV format named result.csv:*

*LastName1,FirstName1*

*LastName2,FirstName2*

*...*

*LastNameN,FirstNameN*

*=end*

**require** 'csv'

**i** = 0

**arr1** = []

**arr2** = []

**arr3** = []

**Dir.glob**(ARGV[0]+"\*.rb") **do** |first\_folder|

**name** = first\_folder.split('/').last.split('.').first.split('\_')

**if** **name**.length == 3

**if** **name**[1].to\_s.length == 5

**arr1**[**i**] = []

**arr**[**i**][0] = **name**[0]

**arr**[**i**][1] = **name**[1]

**i**+=1

```

        end
    end
end
i = 0
Dir.glob(ARGV[1]+"*.rb") do |second_folder|
    name = second_folder.split('/').last.split('.').first.split('_')

    if name.length == 3
        if name[1].to_s.length == 5
            arr1[i] = []
            arr[i][0] = name_1[0]
            arr[i][1] = name_1[1]
            i+=1
        end
    end
end
i = 0
arr1.each do |compare1|
    arr2.each do |compare2|
        if compare2 == compare1
            arr3[i] = compare1
            i+=1
        end
    end
end
sort = arr3.sort_by{|asd| asd[1]}
CSV.open("students.csv", "w") do |csv|
    sort.each do |element|
        csv << element
    end
end
End

```

### 3. Dimitar Nestorov:

#### ○ Dimitar's task:

Develop a program named `FirstName_LastName_ClassNumber_0d5526.rb`

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

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

#### ○ Performance

In Dimitar's program there are no code errors, but a few mistakes could be detected. The first of them is the output file name. The other is the name length check. The third one is the way of reversing the output data. And the last is the sort of the array.

Despite all the mistakes, I found Dimitar's code easy-to-understand. So I would value his work with 3 points.

#### ○ Recommendations

The first mistake is in the length check. According to the task, only files with first names length = 10 should be outputted. Dimitar's program is performing that check but for the last names. The other mistake was assumed when creating file to write into. The name should be **results.csv**, not `result.csv` as in the program. The sort can be performed easily by sorting the array in ASC way and then reverse it in the output.

All my recommendations to Dimitar could be summed within three words: read more carefully.



## ○ Code

```
#Develop a program named FirstName_LastName_ClassNumber_Od5526.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 {|e| -e[1]}
CSV.open("result.csv", "w") do |csv|

  array.uniq.each do |e|

    csv << e

  end
end
```

## 4. Dimitar Terziev:

### ○ Dimitar's task:

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
```

### ○ Performance

Dimitar's code works OK. There are parts that are not easy to understand, but generally it is OK. I'm valuing it with 5 points.

### ○ Recommendations

The program was unaccepted by the checker because it says the files have to be in the format `First_Last_digit.rb` and with only 5 **letters in the last name** to be outputted. Dimitar checks whether the last name contains five letters, but he checks the first name for letters only too, which was not assigned. I mean that file could be named `"F1l3_Hough_3.rb"` and the program has to accept it as a valid name.

## ○ Code

=begin

Develop a program named FirstName\_LastName\_ClassNumber\_88db52.rb

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

```

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

```

=end

require 'csv'

arr = []

Dir.glob("#{ARGV[0]}\*.rb\*"){ |file|

file\_str = file.split('/').last

if(file\_str =~ /\A[a-zA-Z]+\\_[a-zA-Z]+\\_\d+\.rb\z/ && file\_str.split('\_')[1].size == 5)

arr.push("#{file\_str.split('\_')[1]} #{file\_str.split('\_').first}")

end

}

CSV.open('result.csv','w'){ |csv|

arr.uniq.sort.each{ |el|

csv << "#{el.split(' ').last} #{el.split(' ').first}".split(' ')

}

}

## 5. Georgi Ivanov:

### ○ Georgi's task:

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
```

### ○ Performance

Georgi's code works perfect and is easy to be read. 5/5.

### ○ Recommendations

**daiba** is interesting decision for a name of a variable :D

## ○ Code

=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
end

```

## 6. Hristo Dachev:

### ○ Hristo's task:

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 `FrstName_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
```

### ○ Performance

Hristo has built good working program. The code he wrote is easy-to-understand and contains smart solutions. I value his job with 5 points.

### ○ Code

```
=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 `FrstName_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
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
end

```

## 7. Ivelin Slavchev:

### ○ Ivelin's task:

Develop a program named `FirstName_LastName_ClassNumber_835552.rb`

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

1.1 if there are other arguments they should be discarded;

2. Find all the files from both folders that are not in the format

`FirstRName_LastName_digits.rb`. If there are duplicates the file must be written only once. If two files are of the same lenth 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`

### ○ Performance

Ivelin's program contains an error, that is being outputted by the compiler as follows:

`Ivelin_Slavchev_10_835552.rb:33:in `block in <main>': undefined local variable or method `digit' for main:Object (NameError)`

`from Ivelin_Slavchev_10_835552.rb:28:in `each'`

`from Ivelin_Slavchev_10_835552.rb:28:in `<main>'`

This error is interrupting the normal work of the program. It is caused by using undeclared variable. Anyways, after correcting it, the compiler accepts the program, but the output is still not satisfying the task's requirements. That's so because the program contains a few logical mistakes. The first of them is into the format checks. Ivelin had performed a few checks but they are not enough for the program to start working as expected. Another logical mistake is into the **digit1** split.

Despite all the mistakes, his code is legible and in my opinion covers most of the task's purposes. I'm valuing his work with 3 points.

### ○ Recommendations

The error could easily be solved by replacing **digit** in line 33 with **digit2**.

The mistakes and their solutions are the following: to gain the required amount of checks, one have to look for the number of elements in the filename. That could be done by first splitting the filename by point and selecting the first part of the returned result, and after that with splitting by lower slash. The size of the returned array is the count of the elements, contained in the filename. Their count have to be 3 (`First_Last_Digit`) for file to be in the correct format.



The last mistake I found into the digit splitting. The digit has to be only the digit in the filename, but in Ivelin's code it isn't. That's because he missed splitting by point and then selecting the first element of the returned array. With that part missed, **digit1** would look like "4.rb" (instead of just "4"), which definitely doesn't satisfy us. After correcting the mentioned mistakes, the program works as it has to.

## ○ Code

=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 FirstName\_LastName\_digits.rb. If there are duplicates the file must be written only once. If two files are of the same length those files should be sorted in ASC order;
3. Calculate the length of their names (including extensions).;
4. Sort the result by length ;
5. Produce a result in CSV format named result.csv:

```

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

=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 != digit2) 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
```

## 8. Ivo Valchev:

## ○ Ivo's task:

Develop a program named `FirstName_LastName_ClassNumber_6c8bd9.rb`

1. you are given two arguments for a folders with files;
- 1.1 if there are other arguments they should be discarded;
2. file names in this folders are in the form `First_Last_digits.rb`;
3. find the students with 5 letters in the first name that are in both folders. A

student is in both folders if it there is a file with the same First and Last Name. Digits might be different;

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

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

## ○ Performance

Running the program via ruby `Ivo_Valchev_11_6c8bd9.rb 6c8bd9/folder1/6c8bd9/folder2/` outputs error as following: `Ivo_Valchev_11_6c8bd9.rb:23: syntax error, unexpected tIDENTIFIER, expecting key word_then or ';' or '\n' ...== 5 and !isNum!=nil hash_fold1.include?(name[0])`

```
...
^
Ivo_Valchev_11_6c8bd9.rb:26: syntax error, unexpected keyword_end, expecting
end
```

-of-input

The code is readable but it is not easy to understand. Having that in mind and all the mistakes, I'm valuing his work with 3 points.

## ○ Recommendations

Coding with having in mind that other people could look at one's code could boost the the production of a team. Efficiency is not always the most important thing, especially when it comes to understandable code.

## ○ Code

=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]}*.rb") 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]}*.rb") 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
end
```

## 9. Kalin Marinov:

### ○ Kalin's task:

Develop a program named `FirstName_LastName_ClassNumber_bce70c.rb`

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

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

### ○ Performance

Kalin's program has the following error: `Kalin_Marinov_14_12_bce70c.rb:30: syntax error, unexpected keyword_do_block`

`Kalin_Marinov_14_12_bce70c.rb:33: syntax error, unexpected keyword_end, expectin`

g end-of-input. It is caused because the created value is specified before **do** statement. After correcting it, I found a few mistakes. Kalin is not performing any format checks and actions. So he would never know if the submitted file contains three parts (as specified in format), or not. Excluding that, his program is working and his code is plain and readable. I would value his job with 3/5 points.

### ○ Recommendations

The error could be easily fixed by changing the places of **do** and **|key|** in line 30.

For correcting the mistakes, I would recommend Kalin to add some splits and checks into the `Dir.glob` loop.

## ○ Code

# 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
```

```
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 do |key|
    csv << key
  end
end
```

## 10. Kamena Dacheva:

### ○ Kamena's task:

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
```

### ○ Performance

Kamena's program is running without any problems. Her code is legible and easy-to-understand. I would rate her program 5/5.

## ○ Code

=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}/*.rb") 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
end
```

## 11. Kristina Pironkova:

### ○ Kristina's task:

Develop a program named `FirstName_LastName_ClassNumber_890ba0.rb`

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

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

### ○ Performance

Kristina's program is basically working, but with some mistakes. The first of them is the name of the document, where the output data is stored. It has to be **result.csv** instead of **results.csv**.

The second problem I noticed is with the sorting. Her code sorts the data as expected, except the first result.

And at last, she does not perform any data validation, so `"Tenwordfil_Ename.rb"` is accepted as a correct name.

Her code is easy-to-understand and readable, but there are mistakes that affect on the work of the program and on the expected results. Having all that in mind, I would grade her with 3 points.

### ○ Recommendations

My first recommendation is changing the output filename from **result.csv** to **results.csv**.

For the correct sorting I would suggest extending the program a little more and moving the sort method out of the loop. With so sorted array, one has much more freedom of action.

And as a final statement, the data validation should be consisted of count-checking statement (in order to check if the filename could be separated in three parts), digit check (for the task) and extension check.



## ○ Code

```
=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|
```

```

    results.sort.each do |first,last|

```

```

        csv << [last,first]

```

```
    end
```

```
end
```

## 12. Lubomir Yankov:

### ○ Lubomir's task:

Develop a program named `FirstName_LastName_ClassNumber_650c0b.rb`

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

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

### ○ Performance

Lubomir's program is basically working. It doesn't have any syntax errors in the code. It creates a CSV but the file is empty and have a wrong name- `results.csv` not `result.csv`. . He is using only one `Dir.glob` although the programs require two of them. Anyway, the code is very readable so in my opinion he deserves 3 points.

### ○ Recommendations

He should change the output filename from **`result.csv`** to **`results.csv`**.

The program should be extended and pre developed. It will be great if the task was written in the beginning.

## ○ Code

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

array = []
count = 0

Dir.glob(ARGV[0] + "**").each do |file|
  ch_count = 0
  file_name = file.split("/").last.split("")

  file_name.each do |ch|

    if is_numeric(ch)

      ch_count += 1

    end

  end

  if ch_count == 9
    len = file_name.length
    array[count] = []
    array[count][0] = file_name
    array[count][1] = len/2.round
    count += 1
  end

end

array = array.sort_by {|el| el[0]}
CSV.open("results.csv", "w") do |csv|

  array.each do |element|

    csv << element

  end

end
```

### 13. Marian Belchev:

#### ○ Marian's task:

Develop a program named `FirstName_LastName_ClassNumber_ad26e0.rb`

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

```

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

```

#### ○ Performance

Marian's program is basically working. It doesn't have any syntax errors in the code. It creates a CSV but the file is practically empty and have a wrong name- `results.csv` not `result.csv`. . He is using two Hashs which for me is the harder why for doing the job. There should be a mistake in the outputting of the CSV. The code is readable but the last few rows are mess and it1s hard to be understood so in my opinion he deserves 3 points.

#### ○ Recommendations

He should change the output filename from **result.csv** to **results.csv**.

The program is long enough but should be pre developed so as to do good CSV.

## ○ Code

```
=begin
```

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

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

```

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

require 'csv'

hash1 = Hash.new
hash2 = Hash.new

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

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

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

CSV.open("results.csv", "w") do |csv|
  hash2.sort.each do |key, value|
    if !hash1.has_key?(key) && !hash1.has_value?(value.split(".").first) &&
!hash1.has_value?(value.split(".").last.to_i)
      csv << [key,value.gsub('.',',')]
    end
    if hash1.has_key?(key) && !hash1.has_value?(value.split(".").first) &&
!hash1.has_value?(value.split(".").last.to_i)
      csv << [key,value.gsub('.',',')]
    end
  end
end
end
```

## 14. Morreti Georgiev :

- Morreti's task:

=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

- Performance

Moretti's program haven't got any syntax errors. It works well, but the CSV that it creates is empty. I tried to solve the problem but it needs to many changes which will make the code too different from the original. Anyway the program is simple and quite readable so I will give Morreti 4 points.

- Recommendations

The program can be written with some changes in the beginning and all the other parts will work fine after that. Also the final part with the output can be made in more complex but secure way.

## ○ Code

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

## 15. Momchil Angelov:

### ○ Momchil's task:

=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

### ○ Performance

Momchil's program gave the mistake:

Momchil\_Angelov\_18\_d8aa65.rb:44:in `block in <main>': undefined local variable or method `short' for main:Object (NameError)

from Momchil\_Angelov\_18\_d8aa65.rb:40:in `glob'

from Momchil\_Angelov\_18\_d8aa65.rb:40:in `<main>'

This error is interrupting the normal work of the program. It is caused by using undeclared variable. After correcting the mistake by changing "=" to "==" on two places and changing "short" to "short2" the program started to work but still doesn't doing the task. There were no CSV file creating. The code is written readable so I will give 3 points to Momchil.

### ○ Recommendations

The program is long enough but should be pre developed so as to do good CSV. Next time he should look more careful over his code and find a way to solve the problem with the CSV.



## ○ Code

=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 FirstName\_LastName\_digits.rb. If there are duplicates the file must be written only once.
- 2.1 If two files are of the same length those files should be sorted in ASC order;
3. Calculate the length of their names (including extensions).;
4. Sort the result by length ;
5. Produce a result in CSV format named result.csv:

File1,3

File2,4

...

FileN,3

=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= short2.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
  end
end
```

## 16. Nikola Marinov:

### ○ Nikola's task:

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

File1,3

File2,4

...

FileN,3

### ○ Performance

Nikola's program gave the mistake:

Nikola\_Marinov\_20\_add57e.rb:29: syntax error, unexpected '=', expecting keyword\_  
end

```
array(count) = []
      ^
```

Nikola\_Marinov\_20\_add57e.rb:43: syntax error, unexpected '=', expecting keyword\_  
end

```
array(count) = []
      ^
```

Nikola\_Marinov\_20\_add57e.rb:49: syntax error, unexpected '}'

Nikola\_Marinov\_20\_add57e.rb:55: syntax error, unexpected end-of-input, expecting  
'}'

After debugging the errors the program started to work properly and created a CSV file with stuff into it but the whole idea of its algorithm and the work that it done has nothing common with the requirements of the task so it doesn't fulfill the exercise. The code is readable so I will give Nikola 2 points.

## ○ Recommendations

Next time you better read what exactly the condition says. Don not copy/paste a homework task which looks like but is not like the program that you have to write. Take a good look over your code to avoid mistakes and create a suitable loop.

## ○ Code

=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 smallest number;
4. Sort the result by File name ;
5. Produce a result in CSV format named result.csv:

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

=end

require '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.length != 3 && !is\_numeric(name[2])

    array(count) = []

    array(count) [0]=full\_name

    array(count)[1]= full\_name.to\_s.length

    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.length != 3 && !is\_numeric(name[2])

    array(count) = []

    array(count) [0]=full\_name

    array(count)[1]= full\_name.to\_s.length

    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

## 17. Petko Bozhinov:

### ○ Petko's task:

# Develop a program named `FirstName_LastName_ClassNumber_954dc6.rb`

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

# 1.1 if there are other arguments they should be discarded;

# 2. file names in this folders are in the form `First_Last_digits.rb`;

# 3. find the students with 5 letters in the first name that are in both folders. A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;

# 4. Sort the result by Last name ;

# 5. Produce a result in CSV format named `result.csv`:

#      `LastName1,FirstName1`

#      `LastName2,FirstName2`

#      `...`

#      `LastNameN,FirstNameN`

### ○ Performance

Petko`s program works just fine, it doesn`t have any syntax errors and creates a CSV file with content in it, which fulfill all the requirements of the task except one. It reads all file instead this one with extension “.rb”. The program is easy readable so I will give Pepo 4 points for it.

### ○ Recommendations

Next time you better read what exactly the condition is. If you were done the whole exercise putting a look that checks for “.rb” extension now you would have 5 points.

## ○ Code

```
# 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
```

## 18. Radoslav\_Kostadinov:

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
```

### ○ Performance

Radoslav's program works without syntax errors and other issues ,creates a CSV file but without any content in it. The code is readable but the logic error is hard to be found. I gave Radoslav 3 points.

### ○ Recommendations

Next time look over your loom more carefully and try to uncomplicated the code.



## ○ Code

=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
```

## 19. Simeon Shopkin:

- Simeon's task:

Develop a program named `FirstName_LastName_ClassNumber_56a835.rb`

1. you are given two arguments for a folders with files;  
1.1 if there are other arguments they should be discarded;
2. Find all the files from both folders that are not in the format `FirsrName_LastName_digit.rb`. If there are duplicates the file must be written only once. If two files are of the same 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

- Performance

Simeon`s program gave the error:

`Simeon_Shopkin_23_56a835.rb:26:in `block (2 levels) in <main>': undefined method `split' for ["BabaTi", "1"]:Array (NoMethodError)`

from `Simeon_Shopkin_23_56a835.rb:22:in `glob'`

from `Simeon_Shopkin_23_56a835.rb:22:in `block in <main>'`

from `Simeon_Shopkin_23_56a835.rb:21:in `glob'`

from `Simeon_Shopkin_23_56a835.rb:21:in `<main>'`

after a test with folders with files. I can't fix the error . The code is written readable but some kind of messy. I gave Simeon 3 ponit.

- Recommendations

Change the code to be less messy now it has a construction in type "something innto something", and it`s hard to be fixed.

## ○ Code

=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 FirstName\_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

## 20. Stanimir Bogdanov:

### ○ Stanimir's task:

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
```

### ○ Performance

Stanimir's program work without error in the syntax. It creates a CSV file which is empty. There probably is a mistake carried in the CSV creating loop. The code is easy understandable and i`am sure that with some little changes it will work fine. I give Stanimir 4 points.

### ○ Recommendations

Look carefully over your outputting and try to make it in an easy way.

## ○ Code

```
# 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
```

## 21. Stanislav Gospodinov:

### ○ Stanislav's task:

Develop a program named `FirstName_LastName_ClassNumber_b36abb.rb`

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

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

### ○ Performance

Stanlav's program works without error in the syntax. It creates a CSV file which is empty, caused by an error in the logic. Also the CSV should be named "result.csv" not "results.csv". The code is simple and easy readable, so I gave him 3 points.

## Recommendations

The program could be written in an easy way, it can be fixed without any problems.

## ○ Code

```
=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
```



## 22. Stanislav Valkanov:

### ○ Stanislav's task:

Develop a program named `FirstName_LastName_ClassNumber_4482c1.rb`

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

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

### ○ Performance

Stanlav's program works without error in the syntax. It creates a CSV file which is has content but the results are not the expected ones. The reason is because Stanislav has a mistake in one of the loops which can be resolve by changing `"f (last.length == 5)&&(short_name.split("_").size == 3)"` to `"f (short_name.split("_").size == 3)&&(last.length == 5)"`. The code is readable so I give him 4 ponts.

## Recommendations

Take attention over the logic in the loops to avoid stupid errors.

## ○ Code

#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
```

## 23. Tihomir Lidanski:

### ○ Tihomir's task:

Develop a program named `FirstName_LastName_ClassNumber_dafd44.rb`

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

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

### ○ Performance

Tihomir didn't finish his code in time therefore it doesn't compile and does not produce a CSV file. Given the circumstances I'm grading him with 1.

### ○ Recommendations

Perhaps more practice and reading specified literature would help with future projects.

## ○ Code

```
require 'csv'
```

```
Dir.glob(ARGV[0] + "/*.") do |file|  
  name = file.split("/").last.split(".")
```

```
Dir.glob(ARGV[1] + "/*.") do |file|
```

```
  puts name.length % 2.round()
```

```
end  
end
```

```
CSV.open("result.csv", "w") do |csv|
```

```
End
```

## 24. Veselin Dechev:

### ○ Veselin's task:

Develop a program named `FirstName_LastName_ClassNumber_5f1c22.rb`

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

```

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

```

### ○ Performance

Upon running Veselin's program it returns the following error: "undefined method '+' for nil: NilClass (NoMethodError)". The program could be fixed easily, his code is simple and readable and I'm giving him 3 points.

### ○ Recommendations

Perhaps looking into the `Dir.glob` loop would help Veselin as well as naming the file correctly.

## ○ Code

```
require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*.rb").each do |first|
  name1 = first.split("/").last.capitalize
  first_name = name1.split("_").first.capitalize
  last_name = name1.split("_", 2).last.split('_').first.capitalize
  Dir.glob(ARGV[1] + "*.rb").each do |second|
    name2 = second.split("/").last.capitalize
    if (name1 == name2)
      result.compare_by_identity
      result[first_name] = last_name
    end
  end
end
end
CSV.open("result.csv", "w") do |csv|
  result.sort_by{|k, v| k}.each do |element|
    csv << element
  end
end
```

# Class B

## 25. Borislav Stratev:

### ○ Borislav's task:

Develop a program named `FirstName_LastName_ClassNumber_a65be5.rb`

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

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

### ○ Performance

Borislav's code does not return any errors after compiling. I'd give his code 4 points.

### ○ Recommendations

What leads me to believe his program isn't working is the name of the CSV file. In his task it says naming the file `"result.csv"`, instead he named it `"results.csv"`



## ○ Code

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

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

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

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

## 26. David Georgiev:

### ○ David's task:

Develop a program named `FirstName_LastName_ClassNumber_1eea4f.rb`

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

```

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

```

### ○ Performance

David's code has these errors: "undefined method `length' for nil:NilClass (NoMethodError) from ../results/b/results/David\_Georgiev\_12\_1eea4f.rb:17:in `glob' from ../results/b/results/David\_Georgiev\_12\_1eea4f.rb:17:in `'". The code is simple, readable yet it doesn't work, so I'll grade him with 2.

### ○ Recommendations

Looking into the `Dir.glob` loop and trying to fix it should be enough.

## ○ Code

```
require 'csv'

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

  name = current_file.split('/').last.split(/_/)
  if name[1].length == 5
    if not students_names.include?("#{name[1]}", "#{name[0]}") then
      students_names << (["#{name[1]}", "#{name[0]}"])
    end
  end
end

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

## 27. Iliyan Germanov:

### ○ Iliyan's task:

Develop a program named `FirstName_LastName_ClassNumber_f8b0d9.rb`

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

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

### ○ Performance

In Iliyan's case the program runs with no errors and produces a CSV file however it does not fulfill the task. Giving him 3 points.

### ○ Recommendations

I found Iliyan's code a bit hard to read and quite longer than most. He should keep in mind that sometimes a less complex code could be a better solution to a problem.

## ○ Code

```

require 'csv'
results = Hash.new
results.compare_by_identity
def is_number(str)
  str[/[0-9]+/] == str
end
Dir.glob("#{ARGV[0]}/*.rb") do |path1|
  filename1 = path1.split(/\/\//).last
  if filename1.count("_") == 2
    firstname1 = filename1.split("_").first
    lastname1 = filename1.split("_")[1]
    digit1 = filename1.split("_")[2].split(".").first
    if is_number(digit1)
      flag = 0
      Dir.glob("#{ARGV[1]}/*.rb") do |path2|
        filename2 = path2.split(/\/\//).last
        if filename2.count("_") == 2
          digit2 = filename2.split("_")[2].split(".").first
          if is_number(digit2)
            name1 = firstname1 + lastname1
            name2 = filename2.split("_").first +
              filename2.split("_")[1]

            if name1 == name2
              flag = 1
              break
            end
          end
        end
      end
    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
end

```

## 28. Lili Kokalova:

### ○ Lili's task:

Develop a program named `FirstName_LastName_ClassNumber_e0ea9c.rb`

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

```

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

```

### ○ Performance

Lili's code didn't have any errors although the CSV file that's been output doesn't seem to fulfill the task. Grading her with 3 points.

### ○ Recommendations

Changing the block of code that checks whether a student is only in the second folder could be the solution.

## ○ Code

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

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

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

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

end
```

## 29. Nikolay Mihailov:

### ○ Nikolay's task:

Develop a program named `FirstName_LastName_ClassNumber_f70059.rb`

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

File1,3

File2,4

...

FileN,3

### ○ Performance

Upon compiling his code returns several errors: "in `block in <main>': undefined method `each' for \"F10letter3\_L10letter4.rb\":String (NoMethodError) from ../results/b/results/Nikolay\_Mihailov\_25\_f70059.rb:18:in `glob' from ../results/b/results/Nikolay\_Mihailov\_25\_f70059.rb:18:in `<main>'". It doesn't seem a quick fix would be possible. I would give him 2 points.

### ○ Recommendations

I recommend looking into the task's algorithm and trying to solve the problem again.



## ○ Code

```
require 'csv'
hash = Hash.new
count = 0

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

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

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

        c = first[i].chr
        if element == 0 || element == 1 || element == 2 || element == 3 || element == 4 ||
element == 5 || element == 6 || element == 7 || element == 8 || element == 9
            count +=1
        end
    end
    puts count
end

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

end

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

### 30. Stanislav Iliev:

#### ○ Stanislav's task:

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
```

#### ○ Performance

Stanislav's code compiles and produces a CSV file, but it doesn't seem to solve the task. Code isn't as simple as most of the other programs. Giving Stanislav 3 points.

#### ○ Recommendations

Changing the code in the loops could be what he's looking for.

## ○ Code

```

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

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

```

## 31. Stefan Iliev:

### ○ Stefan's task:

Develop a program named `FirstName_LastName_ClassNumber_d77aee.rb`

1. you are given two arguments for a folders with files;
- 1.1 if there are other arguments they should be discarded;
2. Find all the files from both folders that are not in the format `FirstRName_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
```

### ○ Performance

The code compiles without any errors but it does not produce a CSV file due to naming it “results.csv” instead of “result.csv”. It’s readable yet a bit complex so I’d give him 4 points.

### ○ Recommendations

Changing the name of the CSV file to be produced.

## ○ Code

```
require 'csv'

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

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

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

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

puts names_hash

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

## 32. Valentin Varbanov:

### ○ Valentin's task:

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
```

### ○ Performance

The code doesn't compile. Returns the following errors: "in `block in <main>': undefined local variable or method `name' for main:Object (NameError) from ../results/b/results/Valentin\_Varbanov\_4\_041472.rb:44:in `each' from ../results/b/results/Valentin\_Varbanov\_4\_041472.rb:44:in `<main>'". A quick fix can't seem to be applied. I'd grade his work with 2 points.

### ○ Recommendations

Simplifying the code a little bit as well as checking the loops for the errors pointed out.

## ○ Code

```

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
end

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

```

### 33. Veselina Kolova:

#### ○ Veselina's task:

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
```

#### ○ Performance

Veselina's program prints several errors when an attempt for compiling is performed.

Here they are:

```
Veselina.rb:23: syntax error, unexpected tIDENTIFIER, expecting keyword_then or
';' or '\n'
```

```
...le.extname(text_file) text_file.include?(".rb") && text_file...
...                                ^
```

```
Veselina.rb:23: syntax error, unexpected tCONSTANT, expecting keyword_end
...t(/./).first.to_i.is_a Integer then
...                                ^
```

```
Veselina.rb:31: syntax error, unexpected keyword_end, expecting end-of-input
```

They are not tough to be resolved. Anyways, in the code that Veselina has written is almost impossible to find what's wrong.

Also, when I fixed the errors, the output is not satisfying the task. For example, `First_Second_Last.rb` is accepted as a valid formatted name despite it doesn't have to be.

And at last, the output format is not the one requested into the task.

I think that Vesselina has not been thinking about future contributors while developing, so according to my opinion, the program, she developed, is not working as exoected and is not easy to understand. She deserves no more than 2 points.



## ○ Recommendations

In order to make the code easy readable, I would recommend the one-statement-per-line coding style (I just came to that word, but I think you got me) where impossible to write the statement in one line only. It is making the code better-looking and way easier for debugging.

Also, in my opinion it is much easier to have variables declared before the statements, because in this way one has the variable and its value right in front of the sight.

Another recommendation is checking the input result better. In the so provided code, the result is practically being checked for only the second and the last element of the splitted string which could not always satisfy the task.

## ○ Code

```
=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) == ".rb" && text_file.split("/").last.split(".").first.to_i.is_a? Integer then
        if (text_file.split("/").last.split("_").length == 3) then
            text_file = text_file.split("/").last
            if (text_file.split("_")[1].length == 5) then
                people[text_file.split("_")[1]] = text_file.split("_")[0]
            end
        end
    end
end

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

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

## 34. Vladimir Yordanov:

### ○ Vladimir's task:

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
```

### ○ Performance

Vladimir's program is basically working. I detected a few problems that are making the code incorrect. The first of them is the output file. It has to be **result.csv** instead of **results.csv**. The other problems are linked with the little amount of filename checks. For example, "**File\_NameF\_OrHere.rb**" is accepted as a valid filename. But that's not tough error, having in mind that it could be resolved with a few more splits. Vladimir's code is readable and easy-to-understand. That's why I'm going to value his work with 3 points.

### ○ Recommendations

Vladimir needs to check the filename a few more times in order to resolve all the problems in his code.

## ○ Code

# 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|
  names.to_a.each do |element|
    csv << element
  end
end
```