



Технологично Училище  
“Електронни системи” към  
ТУ-СОФИЯ

Радослав Костадинов  
Борислав Русинов  
Кристина Пиронкова

---

# **Доклад за резултатите от контролна работа № 1**

## **“ErrorBusters”**

Октомври 2014

## **1.Purpose of the report**

The aim of this report is to summarise the results from the first exam in Software Engineering Course in ELSYS. The results provided below are from classes class 11"A" and class 11"B".

## **2.Statistics**

The amount of the submitted programs is 24 from 11a class and 10 from 11b class which equals to 34 submitted programs out of 58 students. Unfortunately, only 5 were correct. The mistaken programs have different types of errors, but we can categorize them to :

**1.Wrong File Name**

**2.Wrong Name of the '.csv' file**

**3.Wrong sorting**

**4.Mistyped keywords**

**5.Unnecessary printing**

**6.Wrong Whole Program**

## **3.Recommendations for fixing the errors**

- It would be nice to emphasize to the students that the Automatic Program Checker needs exact names of files
- It would be nice if they were using `key_words` which they know
- It would be nice if students had prepared appropriate fixtures with which they can test their programs
- It would be nice if students have commented their tasks in the beginning of the file, because it is difficult to understand what are the errors if we do not know what problem students have to solve

# Appendixes

## A CLASS

### 1. Borislav Rusinov:

#### Source 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
```

### **Errors:**

Wrong file name. Writing in “results.csv” instead of “result.csv”. No checking for right format of files.  
Does not reverse, only sorting .

### **Solutions:**

No need of “array.reverse!”. It can be fixed by using hash. If hash was used it would be easier to sort and reverse the names of the files.

### **Ranking:**

**3/5**

## **2.Veselin Dechev:**

### **Source code:**

```
=begin
```

Develop a program named FirstName\_LastName\_ClassNumber\_5f1c22.rb

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

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

```
=end
```

```

require 'csv'
result = Hash.new
Dir.glob(ARGV[0] + "*.rb").each do |first|
  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

```

### Errors:

Wrong file name. Does not sort by last name. Pushes only the files that are in both folders.

### Solutions:

“

```

if (name1 == name2)
  result.compare_by_identity
  result[first_name] = last_name
end

```

” He can change this with:

“

```

if (name1 != name2)
  result.compare_by_identity
  result[first_name] = last_name
end

if !name2.include?(name1)
  result[first_name]=last_name
end

```

end

”->Here he pushes only the files from first folder

```

“
if (name1 == name2)
    result.compare_by_identity
    result[first_name] = last_name
end
” -> Here he must set last_name to key and first_name to value. After that he can easily sort the
hash by key.
“

CSV.open("result.csv", "w") do |csv|
result.sort_by{|k, v| k}.each do |element|
    csv << element
    end
end
”-> Here he can say “File.open(“result.csv”, “w”) do |csv|
    result.sort_by{|k, v|.k}.each do |element|
        csv.puts("#{element[1]},#{element[0]}”)
    end
end” which will print Last_Name,First_Name

```

**Ranking:**

**1/5**

### **3.Georgi Ivanov**

#### **Source 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  
end

**Errors:**  
None.

**Solutions:**

There aren't any errors.

**Ranking:**  
**5/5**

## **4.Denis Trenchev**

### **Source 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

```

```

    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

```

## 5.Dimitar Terziev

### Source 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(' ')
}

```

### Errors:

There is a expression that is checking if there are only letters in the name of the file “file\_str=~\A[a-zA-Z]+\\_[a-zA-Z]+\\_\d+\.rb\z/” which is wrong.

### Solution:

By deleting the expression”file\_str=~\A[a-zA-Z]+\\_[a-zA-Z]+\\_\d+\.rb\z/”.

### Ranking:

**4/5**

## 6.Dimitar Nestorov

### Source code :

```

#Develop a program named FirstName_LastName_ClassNumber_0d5526.rb
#
#1. you are given an argument for a folder with files;
#1.1 if there are other arguments they should be discarded
#2. file names in this folder are in the form First_Last_digits.rb;
#3. find all the students that have 10 letters in their first name;
#4. Sort the result by Last Name DESC.
#5. Produce a result in CSV format named result.csv:
#
#           FirstName1,LastName1
#           FirstName2,LastName2
#           ...
#           FirstNameN,LastNameN

```

```

require 'csv'
def is_numeric(o)
  true if Integer(o) rescue false
end
array = []
count = 0
Dir.glob(ARGV[0] + "*.rb") do |file|
  name = file.split("/").last.split(".").first.split("_")

  name[0] = name[0].to_s
  name[0] = name[0].capitalize

  name[1] = name[1].to_s
  name[1] = name[1].capitalize

  if name.size == 3 && is_numeric(name[2])
    if name[1].length == 10

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

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

  array.uniq.each do |e|

    csv << e

  end

end
end

```

### Errors:

```

"#{name[1].to_s}"
"array = array.sort_by {|e| -e[1]}"

```

```

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

  array.uniq.each do |e|

    csv << e

  end

end
"

```

### **Solution:**

"#{name[1].to\_s}"->There mustn't be space

"array = array.sort\_by {|e| -e[1]}"-> This " - " is not needed and the program crashes because of it. He must say "...e[1]}.reverse" to reverse the by last\_name

```

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

  array.uniq.each do |e|

    csv << e

  end

end

```

"->Here he does not prints correctly the files in "result.csv"(he prints them First\_Name,Last\_Name). He must change it with

```

"CSV.open("result.csv", "w") do |csv|
  array.uniq.each do |e|
    csv<< [e[0],e[1]]
  end
end
"

```

### **Ranking:**

## 7.Ivelin Slavchev

### Source 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 FirsrName_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
=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)
  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
end

```

### Errors:

In the second Dir.glob there is a mistyped variable which is not existing -  
“(digit2.to\_i.to\_s != digit) ”

### Solution:

We should rename the variable in that condition- “(digit2.to\_i.to\_s != digit) ” from  
“digit” to “digit2”

### Ranking:

## 8.Ivo Valchev

### Source 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 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

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

```

### Errors:

```

"...and!hash_fold2.include?(name[0])
"

```

Prints nothing

### Solution:

```

"and!hash_fold2.include?(name[0])
"-> There must be space between "and" and "!hash_fold2..."

```

### Ranking:

**3/5**

## 9.Kalin Marinov

### Source code:

```

#==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
end

```

### **Errors:**

Wrong name file.  
Does not split the files in right way.  
Wrong printing.

### **Solution:**

File must be “Kalin\_Marinov\_12\_bce70c.rb”

For correctly splitting of file he must make variable last\_name  
and then : “last\_name=name.split('\_')[1] ” “short\_name=name.split('\_')[0]”. After

that he must write "...hash[short\_name] = last\_name" to set key="short\_name" and value="last\_name"

```
    "hash.each |key| do
      csv << key
    end
  "-> Must be changed with "hash.each do |value|
    csv << value
  end
  " -> This will print
  "First_Name,Last_Name"
```

**Ranking:**

**2/5**

## **10.Kamena Dacheva**

### **Source 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}/*.txt") 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

```

### **Errors:**

`"student.sort_by{|k, v| v}"`

### **Solution:**

Writing this `"student.sort_by{|k, v| v}"` she sorts by last name which is wrong. She must sort by first\_name-> `"student.sort_by{|k,v| k}"`

### **Ranking:**

**4/5**

## **11.Kristina Pironkova**

### **Source 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

end

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

        csv << [last,first]

    end

end
end

```

### **Errors:**

Writing into “results.csv” instead of “result.csv”  
 Sorting in ASC instead of DESC

### **Ranking:**

**4/5**

## 12.Lubomir Yankov

### Source 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
```

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

  array.each do |element|

    csv << element

  end

end
```

### **Errors:**

Writing into “results.csv” instead of “result.csv”.

There is no variable that stores the full name and when he splits it to characters he passes that splitted variable to the CSV file.

### **Solution:**

Make a variable that stores the full name and when writing into the array he should use it instead the splitted one.

Ranking:

**4/5**

**13.Marian Belchev**

**Source 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

```

### **Errors:**

Not readable code !

### **Solution:**

Not readable code!

### **Ranking:**

**3/5**

## **14.Momchil Angelov**

=begin

Develop a program named FirstName\_LastName\_ClassNumber\_d8aa65.rb

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

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

```
=end
```

```
require 'csv'
```

```
arr1=Array.new
```

```
arr2=Array.new
```

```
arr3=Array.new
```

```
a = ARGV[0]
```

```
b = ARGV[1]
```

```
i=0
```

```
Dir.glob(a + "/*.rb") do |my_text_file1|
```

```
  short= my_text_file1.split('/').last
```

```
  length1 = short.length
```

```
  shorter= short.split('.').first.split('_')
```

```
  first_name=shorter[0]
```

```
  last_name=shorter[1]
```

```
  digits=shorter[2].to_i
```

```

    if !first_name || !last_name || digits=0
      next
    else
      arr1 << ["#{short}" "#{length1}"]
    end
  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 { |e|
  e[1]
}

```

```

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

arr3.each do |element|
  csv << element
end

end

```

## Errors:

### 15. Moreti Georgiev

#### Source 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
end
CSV.open("result.csv", "w") do |csv_file|
  student.sort.each do |key, value|
    csv_file << ["#{key}, #{value}"]
  end
end
end

```

### **Errors:**

Writing files in result.csv with quotes.

### **Solutions:**

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

### **Ranking:**

**4/5**

## **16.Nikola Marinov**

### **Source 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.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

```

### **Errors:**

Wrong using of array type many times. A lot mistyped keywords. There is no check whether there are 7 digits in the name of the file.

### **Solutions:**

The amount of corrections is big.

### **Ranking:**

**1/5**

### **17.Petko Bozhinov**

#### **Source 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]
        end
      end
    end
  end
  i += 1
end
```

```

        i+=1
      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

```

### Errors:

Wrong sorting, pushing in hash.

### Solutions:

```

"...&& file2[0].to_s.length==5"
"...output[i][0] = file[1]
  output[i][1] = file[0]
"
"
...element[0]}
"

```

### Ranking:

**4/5**

**18.Radoslav Kostadinov**

### Source 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
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
end
end
end

```

### Errors:

Wrong checking if the file is only in the second folder.

### **Solution:**

```
“  
    file2.delete_if{|x| file1.include?(x)}  
    CSV.open("result.csv", "w") do |csv|  
    file2.sort.each do |push|  
    csv << push  
    end  
end”
```

### **Ranking:**

**3/5**

## **19.Simeon Shopkin**

### **Source 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 FirstsName\_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

end

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

```

### Errors:

Too many errors. Second\_file not splitted. Not needed if-construction...

### Solutions:

There is no quick solution to that program.

### Ranking:

2/5

## 20.Stanimir Bogdanov

### Source code:

```
=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
```

```
end  
end
```

**Errors:**

There aren't any errors.

**Solutions:**

There is no need for any solution.

**Ranking:**

**5/5**

**21.Stanislav Gospodinov**

**Source 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

```

### **Errors:**

Writing into “results.csv” instead of “result.csv”

### **Solution:**

Rename it to “result.csv”

### **Ranking:**

**5/5**

## **22.Stanislav Vulkanov**

### **Source 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
```

### Errors:

No errors

### Ranking

**5/5**

**23.Tihomir Lidanski**

**Source code:**

#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
```

```
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
```

### **Errors:**

Too many. There is almost nothing written.

### **Solutions:**

There is no quick solution.

### **Ranking:**

**0/5**

## **24.Hristo Dachev**

### **Source 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 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'

hash = Hash.new

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

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

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

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

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

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

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

```
        csv << ["#{name}", "#{length}"]
    end
end
```

### **Errors:**

There are no errors

### **Ranking:**

**5/5**

## **B CLASS**

### **1.Borislav Stratev**

#### **Source code:**

#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|
  a = h.sort
  a.each do |element|
    csv << element
  end
end

```

### **Errors:**

Wrong pushing into csv file. He pushes into  
 “results.csv” instead of “result.csv”

**Solution:**

Rename it

**Ranking:**

**5/5**

**2.Valentin Varbanov**

**Source code:**

=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

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

    end

    studentcsv.push(name[1], name[2])
end

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

```

**Errors:**  
**Too many errors.**  
**Solutions:**

**Too many errors.**

**Ranking:**

**1/5**

### **3.Veselina Kolova**

#### **Source 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) text_file.include?(".rb") &&
text_file.split(/_/.last.split(/\./).first.to_i.is_a Integer then
        if (text_file.split("/").last.split("_").length == 3) then
```

```

        text_file = text_file.split("/").last
        if (text_file.split("_")[1].length == 5) then
            people[text_file.split("_")[1]] = text_file.split("_")[0]
        end
    end
end
end

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

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

```

### **Errors:**

**Unclear what she wants to do.**

### **Solutions:**

**There is no quick solution.**

### **Ranking:**

**2/5**

## **4.Valentin Yordanov**

Source 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

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
```

**Errors:**

Wrong csv file.

**Solution:**

He must write into “result.csv” instead of “results.csv”

**Ranking:**

**5/5**

**5.David Georgiev**

**Source code:**

#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].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
end

```

### Errors:

Not needed space here “csv << ["#{first}", "#{last}"]”.

### Solution:

By removing it.

### Ranking:

**5/5**

**6.Iliyan Germanov**

**Source code:**

=begin

Develop a program named FirstName\_LastName\_ClassNumber\_f8b0d9.rb

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

LastName1,FirstName1

LastName2,FirstName2

...

LastNameN,FirstNameN

=end

require 'csv'

results = Hash.new

results.compare\_by\_identity

def is\_number(str)

str[/[0-9]+/] == str

end

Dir.glob("#{ARGV[0]}/\*.rb") do |path1|

filename1 = path1.split(/\/).last

if filename1.count("\_") == 2

firstname1 = filename1.split("\_").first

lastname1 = filename1.split("\_")[1]

digit1 = filename1.split("\_")[2].split(".").first

if is\_number(digit1)

flag = 0

Dir.glob("#{ARGV[1]}/\*.rb") do |path2|

```

        filename2 = path2.split(/\/).last
        if filename2.count("_") == 2
            digit2 = filename2.split("_")[2].split(".").first
            if is_number(digit2)
                name1 = firstname1 + lastname1
                name2 = filename2.split("_").first +
filename2.split("_")[1]
                if name1 == name2
                    flag = 1
                    break
                end
            end
        end
    end
end
if flag == 0
    results[lastname1] = firstname1
end
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

```

### **Errors:**

No checking for wrong file formats.

### **Solution:**

“

```

unless (path1.split('/').last =~
/^[a-zA-Z0-9]+_[a-zA-Z0-9]+_[0-9]+.rb$/).nil?

```



“ -> He must add this after Dir.glob... to check if the file is in the right form

**Ranking:**

**5/5**

## **7.Lili Kokalova**

### **Source code:**

=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
  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
  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 && ln!=ln1
        csv << ["#{fn1}", "#{ln1}"]
      end
    end
  end
end
end
end

```

**Errors:**

Program is printing too many times.

**Solution:**

There is no quick solution.

**Ranking:**

**3/5**

**8.Nikolay Mihailov**

**Source code:**

#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
end

```

**Errors:**

Wrong usage of “each” ->”first.each do |element|”

**Solutions:****Ranking:**

**3/5**

**9.Stanislav Iliev****Source code:**

```
#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|

```

```
end  
    end  
end  
end  
csv << [element2]
```

## Errors:

Too many errors. Missing splitting, etc.

### Solutions:

Too many errors.

### Ranking:

**2/5**

## 10.Stefan Iliev

## Source code:

#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**

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

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

```

### Errors:

Too many errors-missing checking of format,  
wrong csv file.

### Solution:

Too many errors.

### Ranking:

$$\frac{2}{5}$$