

# Learning from mistakes

Team: **Happy Satan** ☺ of the **Illuminati** organization

## **Purpose:**

The purpose of this report is to study the mistakes we have made during our first test so we don't have to repeat them again.

## **Main Problems:**

The main problems that people encountered were

- Not committing on time or at all
- Wrong name of the CSV file
- Unable to sort correctly
- Some programs are beyond fixable

## **Recommendations to avoid problems:**

- Create your own fixtures so you can test your program
- Make it so that other people can read the code
- Pay attention to the task

## **Team:**

Dimitar Nestorov  
Georgi Ivanov  
Momchil Angelov

```
1 =begin
2 Develop a program named FirstName_LastName_ClassNumber_6fb3ad.rb
3
4 1. you are given an argument for a folder with files;
5 1.1 if there are other arguments they should be discarded
6 2. file names in this folder are in the form First_Last_digits.rb;
7 3. find all the students that have 10 letters in their first name;
8 4. Sort the result by Last Name DESC.
9 5. Produce a result in CSV format named result.csv:
10
11     FirstName1,LastName1
12     FirstName2,LastName2
13     ...
14     FirstNameN,LastNameN
15
16 =end
17 a=ARGV[0]
18 require 'csv'
19 array=[]
20 Dir.glob("#{a}*.rb") do |my_text_file|
21     name = my_text_file.split("/").last.split(".").first.split("_")
22     if name[1]!=nil && name[0].length==10
23         array << name[0] + "," + name[1]
24     end
25 end
26 array.sort!
27 array.reverse!
28 File.open("results.csv", "w") do |csv|
29     array.each do |arg|
30         csv.puts(arg)
31     end
32 end
33 end
```

The problem here is that Borislav named his CSV “results” instad of “result”.

The solution is to pay attention to what he has to do.

Score: 4/5 would work if CSV name was correct

```

1 =begin
2 Develop a program named FirstName_LastName_ClassNumber_b4c3f5.rb
3
4 1. you are given two arguments for a folders with files;
5 1.1 if there are other arguments they should be discarded;
6 2. file names in this folders are in the form First_Last_digits.rb;
7 3. find the students with 5 letters in the first name that are in both folders.
8 A student is in both folders if it there is a file with the same First and Last Name. Digits might be different;
9 4. Sort the result by Last name ;
10 5. Produce a result in CSV format named result.csv:
11
12     LastName1,FirstName1
13     LastName2,FirstName2
14     ...
15     LastNameN,FirstNameN
16
17 =end
18
19 require 'csv'
20
21 i = 0
22 arr1 = []
23 arr2 = []
24 arr3 = []
25
26 Dir.glob(ARGV[0]+"*.rb") do |first_folder|
27     name = first_folder.split('/').last.split('.').first.split('_')
28
29     if name.length == 3
30         if name[1].to_s.length == 5
31             arr1[i] = []
32             arr[i][0] = name[0]
33             arr[i][1] = name[1]
34             i+=1
35         end
36     end
37 end
38 i = 0
39
40 Dir.glob(ARGV[1]+"*.rb") do |second_folder|
41     name = second_folder.split('/').last.split('.').first.split('_')
42
43     if name.length == 3
44         if name[1].to_s.length == 5
45             arr1[i] = []
46             arr[i][0] = name_1[0]
47             arr[i][1] = name_1[1]
48             i+=1
49         end
50     end
51 end
52 i = 0
53
54 arr1.each do |compare1|
55     arr2.each do |compare2|
56         if compare2 == compare1
57             arr3[i] = compare1
58             i+=1
59         end
60     end
61 end
62
63 sort = arr3.sort_by{|asd| asd[1]}
64 CSV.open("students.csv", "w") do |csv|
65     sort.each do |element|
66         csv << element
67     end
68 end
69 end

```

Where did name\_1 come from? Also a lot of other errors.

Can't find a solution for this one – too many mistakes.

Score: 1/5

```
1 #Develop a program named FirstName_LastName_ClassNumber_0d5526.rb
2 #
3 #1. you are given an argument for a folder with files;
4 #1.1 if there are other arguments they should be discarded
5 #2. file names in this folder are in the form First_Last_digits.rb;
6 #3. find all the students that have 10 letters in their first name;
7 #4. Sort the result by Last Name DESC.
8 #5. Produce a result in CSV format named result.csv:
9 #
10 #     FirstName1,LastName1
11 #     FirstName2,LastName2
12 #     ...
13 #     FirstNameN,LastNameN
14
15
16 require 'csv'
17 def is_numeric(o)
18   true if Integer(o) rescue false
19 end
20 array = []
21 count = 0
22 Dir.glob(ARGV[0] + "/*.rb") do |file|
23   name = file.split("/").last.split(".").first.split("_")
24
25   name[0] = name[0].to_s
26   name[0] = name[0].capitalize
27
28   name[1] = name[1].to_s
29   name[1] = name[1].capitalize
30
31   if name.size == 3 && is_numeric(name[2])
32     if name[1].length == 10
33
34       array[count] = []
35       array[count][0] = name[0].to_s
36       array[count][1] = " #{name[1].to_s}"
37       count += 1
38     end
39   end
40 end
41
42 array = array.sort_by {|e| -e[1]}
43 CSV.open("result.csv", "w") do |csv|
44
45   array.uniq.each do |e|
46
47     csv << e
48
49   end
50 end
51 end
```

Problems with the reversing of the array

Solution: use .reverse

Score: 3/5

```
1 =begin
2 Develop a program named FirstName_LastName_ClassNumber_88db52.rb
3
4 1. you are given an argument for a folder with files;
5 1.1 if there are other arguments they should be discarded
6 2. file names in this folder are in the form First_Last_digits.rb;
7 3. find all the students that have 5 letters in their second name;
8 4. Sort the result by Last Name ASC.
9 5. Produce a result in CSV format named result.csv:
10
11     FirstName1,LastName1
12     FirstName2,LastName2
13     ...
14     FirstNameN,LastNameN
15
16
17 =end
18 require 'csv'
19 arr = []
20 Dir.glob("#{ARGV[0]}*.rb*"){|file|
21     file_str = file.split('/').last
22     if(file_str =~ /\A[a-zA-Z]+\_[a-zA-Z]+\_\d+\.rb\z/ && file_str.split('_')[1].size == 5)
23         arr.push("#{file_str.split('_')[1]} #{file_str.split('_').first}")
24     end
25 }
26 CSV.open('result.csv', 'w'){|csv|
27     arr.uniq.sort.each{|el|
28         csv << "#{el.split(' ').last} #{el.split(' ').first}".split(' ')
29     }
30 }
```

He didn't need to check if the name contains Letters or Numbers

Solution: Delete `if(file_str =~ /\A[a-zA-Z]+\_[a-zA-Z]+\_\d+\.rb\z/`

Score: 3/5

```
1 =begin Develop a program named FirstName_LastName_ClassNumber_871529.rb
2
3 1. you are given an argument for a folder with files;
4 1.1 if there are other arguments they should be discarded
5 2. file names in this folder are in the form First_Last_digits.rb;
6 3. find all the students that have 5 letters in their second name;
7 4. Sort the result by First name DESC.
8 5. Produce a result in CSV format named result.csv:
9
10     FirstName1,LastName1
11     FirstName2,LastName2
12     ...
13     FirstNameN,LastNameN
14 =end
15
16
17 require "csv"
18
19 arr = []
20 i = 0
21
22 Dir.glob(ARGV[0]+"*.rb") do |file|
23     name = file.split('/').last.split('.').first.split('_')
24     firstname = name[0]
25     lastname = name[1]
26     exercise = name[2]
27
28     if firstname == '' || lastname == '' || exercise == ''
29     elsif name.length == 3
30
31         if lastname.length == 5
32             arr[i] = []
33             arr[i][0] = name[0]
34             arr[i][1] = name[1]
35             i+=1
36         end
37     end
38 end
39
40
41 daiba = arr.sort_by{|asd| asd[0]}.reverse!
42 CSV.open("result.csv", "w") do |csv|
43     daiba.each do |element|
44         csv << element
45     end
46 end
```

Problem: It works

Solution: None! It will continue to work.

Score: I r8 8/8 m8. Pls don't h8.

```

1  =begin
2  Develop a program named FirstName_LastName_ClassNumber_4a196f.rb
3
4  1. you are given two arguments for a folders with files;
5  1.1 if there are other arguments they should be discarded;
6  2. Find all the files from both folders that are not in the format FirsrName_LastName_digits.rb. If
   there are duplicates the file must be written only once. If two files are of the same lenght those
   files should be sorted in ASC order;
7  3. Calculate the length of their names (including extensions).;
8  4. Sort the result by lenth ;
9  5. Produce a result in CSV format named result.csv:
10
11      File1,3
12      File2,4
13      ...
14      FileN,3
15  =end
16
17  require 'csv'
18
19  hash = Hash.new
20
21  Dir.glob("#{ARGV[0]}*").each do |path|
22      first_name = path.split("/").last.split("_").first
23      last_name = path.split("/").last.split("_", 2).last.split("_").first
24      digit = path.split("/").last.split("_", 2).last.split("_").last.split(".").first
25      name = path.split("/").last
26
27      if name.include? "_" then counter = name.count "_" end
28
29      if (counter != 2) || (digit.to_i.to_s != digit)
30          l = name.length
31          hash[name] = l
32      end
33  end
34  Dir.glob("#{ARGV[1]}*").each do |path|
35      first_name = path.split("/").last.split("_").first
36      last_name = path.split("/").last.split("_", 2).last.split("_").first
37      digit = path.split("/").last.split("_", 2).last.split("_").last.split(".").first
38
39
40
41      name = path.split("/").last
42      if name.include? "_" then counter = name.count "_" end
43
44      if (counter != 2) || (digit.to_i.to_s != digit)
45          l = name.length
46          hash[name] = l
47      end
48  end
49
50  CSV.open("result.csv", "w") do |csv|
51      hash.sort_by{ |k, v| v}.each do |name, length|
52          csv << ["#{name}", "#{length}"]
53      end
54  end

```

Problem: None

Solution: None needed

Score: 5/5



Ivelin Slavchev

```
1 =begin
2   Develop a program named FirstName_LastName_ClassNumber_835552.rb
3
4   1. you are given two arguments for a folders with files;
5   1.1 if there are other arguments they should be discarded;
6   2. Find all the files from both folders that are not in the format FirsrName_LastName_digits.rb. If there are
   duplicates the file must be written only once. If two files are of the same lenght those files should be
   sorted in ASC order;
7   3. Calculate the length of their names (including extensions).;
8   4. Sort the result by lenth ;
9   5. Produce a result in CSV format named result.csv:
10
11       File1,3
12       File2,4
13       ...
14       FileN,3
15 =end
16
17 require 'csv'
18 result = Hash.new
19 Dir.glob(ARGV[0] + "*").each do |file1|
20   short1 = file1.split("/").last
21   ext1 = short1.split(".").last
22   names1 = short1.split(".").first
23   digit1 = file1.split("_").last
24   if (ext1 != "rb") or (digit1.to_i.to_s != digit1) or (short1.scan("_").count != 2)
25     result[short1] = short1.length
26   end
27 end
28 Dir.glob(ARGV[1] + "*").each do |file2|
29   short2 = file2.split("/").last
30   ext2 = short2.split(".").last
31   names2 = short2.split(".").first
32   digit2 = file2.split("_").last
33   if (ext2 != "rb") or (digit2.to_i.to_s != digit2) or (short2.scan("_").count != 2)
34     result[short2] = short2.length
35   end
36 end
37 result.sort_by{|k, v| v}
38 CSV.open("result.csv", "w") do |csv|
39   result.each do |p|
40     csv << p
41   end
42 end
```

Problem: there are multiple problems with Ivelin's program and one of them is forgetting his variables names

Soltuion: Compile your program to see if it works before committing.

Score: 3/5



```

1 =begin
2 Develop a program named FirstName_LastName_ClassNumber_6c8bd9.rb
3
4 1. you are given two arguments for a folders with files;
5 1.1 if there are other arguments they should be discarded;
6 2. file names in this folders are in the form First_Last_digits.rb;
7 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;
8 4. Sort the result by Last name ;
9 5. Produce a result in CSV format named result.csv:
10
11     LastName1,FirstName1
12     LastName2,FirstName2
13     ...
14     LastNameN,FirstNameN
15 =end
16
17 hash_fold1={}
18 hash_fold2={}
19
20 Dir.glob("#{ARGV[0]}*.rb") do |file|
21     name = file.split("/").last.split(".").first.split("_")
22     isNum = Integer(name[2]) rescue nil
23     if name[0] and name[1] and name[0].length == 5 and !isNum!=nil hash_fold1.include?(name[0])
24         hash_fold1["#{name[1]}"] = "#{name[0]}"
25     end
26 end
27 Dir.glob("#{ARGV[1]}*.rb") do |file|
28     name = file.split("/").last.split(".").first.split("_")
29     isNum = Integer(name[2]) rescue nil
30     if name[0] and name[1] and name[0].length == 5 and !isNum!=nil and !hash_fold2.include?(name[0])
31         hash_fold2["#{name[1]}"] = "#{name[0]}"
32     end
33 end
34 File.open("result.csv", "w") do |csv|
35     hash_fold1.sort.map do |key, value|
36         if (hash_fold1[key]==hash_fold2[key])
37             csv.puts("#{key},#{value}")
38         end
39     end
40 end

```

Problems: Too many

Solution: l2program

Score: 2/5

```

1 #Develop a program named FirstName_LastName_ClassNumber_a65be5.rb
2
3 #1. you are given two arguments for a folders with files;
4 #1.1 if there are other arguments they should be discarded;
5 #2. file names in this folders are in the form First_Last_digits.rb;
6 #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;
7 #4. Sort the result by Last name ;
8 #5. Produce a result in CSV format named result.csv:
9
10 # LastName1,FirstName1
11 # LastName2,FirstName2
12 # ...
13 # LastNameN,FirstNameN
14
15 require 'csv'
16 a = Array.new
17 h = Hash.new
18 Dir.glob("#{ARGV[0]}/*.rb") do |dir_file_name_1|
19   Dir.glob("#{ARGV[1]}/*.rb") do |dir_file_name_2|
20
21     file_name_1 = dir_file_name_1.split(/\/\//).last.to_s
22     file_name_2 = dir_file_name_2.split(/\/\//).last.to_s
23
24     if(file_name_1 != file_name_2)
25       file_name = file_name_1
26       digit = file_name.split(/_/).last.split(/\./).first.to_s
27       first_name = file_name.split(/_/).first.to_s
28       full_first_name = first_name + digit
29       full_first_name = full_first_name.to_s
30       tmp = file_name.split("#{first_name}_")
31       full_last_name = tmp.last.split(/_/).first.to_s + digit
32       full_last_name = full_last_name.to_s
33       h[full_last_name] = full_first_name
34
35     end
36   end
37 end
38
39 CSV.open("results.csv", "w") do |csv|
40   a = h.sort
41   a.each do |element|
42     csv << element
43   end
44 end

```

Problems: Too many

Score: 0/5

```

1      #Develop a program named FirstName_LastName_ClassNumber_1eea4f.rb
2
3      #1. you are given an argument for a folder with files;
4      #1.1 if there are other arguments they should be discarded
5      #2. file names in this folder are in the form First_Last_digits.rb;
6      #3. find all the students that have 5 letters in their second name;
7      #4. Sort the result by Last Name ASC.
8      #5. Produce a result in CSV format named result.csv:
9
10     #      FirstName1,LastName1
11     #      FirstName2,LastName2
12     #      ...
13     #      FirstNameN,LastNameN
14
15     require 'csv'
16     students_names = []
17     Dir.glob("#{ARGV[0]}/*/*/*.rb") do |current_file|
18
19         name = current_file.split('/').last.split(/_/)
20         if name[1].length == 5
21             if not students_names.include?("#{name[1]}", "#{name[0]}") then
22                 students_names << (["#{name[1]}", "#{name[0]}"])
23             end
24         end
25     end
26     CSV.open("result.csv", "w") do |csv|
27         students_names.sort.each do |last, first|
28             csv << ["#{first}", "#{last}"]
29         end
30     end

```

Mark: 3/5

To fix this, I suggest using this if:

```
if !name[0] || !name[1] || name[2].to_i == 0
```

```
next
```

```
else if name[1].length == 5 and so on with his code.
```

This will make his program check first then ask for size, which will fix nilclass error.

His sort is sorting by 1<sup>st</sup> element too, so we have to fix it too.

```

1 =begin
2   Develop a program named FirstName_LastName_ClassNumber_f8b0d9.rb
3
4   1. you are given two arguments for a folders with files;
5   1.1 if there are other arguments they should be discarded;
6   2. file names in this folders are in the form First_Last_digits.rb
7   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;
8   4. Sort the result by Last name ;
9   5. Produce a result in CSV format named result.csv:
10
11   LastName1,FirstName1
12   LastName2,FirstName2
13   ...
14   LastNameN,FirstNameN
15 =end
16
17 require 'csv'
18 results = Hash.new
19 results.compare_by_identity
20 def is_number(str)
21   str[/[0-9]+/] == str
22 end
23 Dir.glob("#{ARGV[0]}/*.rb") do |path1|
24   filename1 = path1.split(/\/\//).last
25   if filename1.count("_") == 2
26     firstname1 = filename1.split("_").first
27     lastname1 = filename1.split("_")[1]
28     digit1 = filename1.split("_")[2].split(".").first
29     if is_number(digit1)
30       flag = 0
31       Dir.glob("#{ARGV[1]}/*.rb") do |path2|
32         filename2 = path2.split(/\/\//).last
33         if filename2.count("_") == 2
34           digit2 = filename2.split("_")[2].split(".").first
35           if is_number(digit2)
36             name1 = firstname1 + lastname1
37             name2 = filename2.split("_").first + filename2.split("_")[1]
38             if name1 == name2
39               flag = 1
40               break
41             end
42           end
43         end
44       end
45       if flag == 0
46         results[lastname1] = firstname1
47       end
48     end
49   end
50 end
51
52 CSV.open("result.csv", "w") do |csv|
53   results.sort_by{|key, val| key}.each do |el|
54     csv << el
55   end
56 end

```

Mark: 2/5

Works, but you can't read easily how the program works.

```

1 =begin
2 Develop a program named FirstName_LastName_ClassNumber_e0ea9c.rb
3
4 1. you are given two arguments for a folders with files;
5 1.1 if there are other arguments they should be discarded;
6 2. file names in this folders are in the form First_Last_digits.rb;
7 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;
8 4. Sort the result by First name ;
9 5. Produce a result in CSV format named result.csv:
10
11     LastName1,FirstName1
12     LastName2,FirstName2
13     ...
14     LastNameN,FirstNameN
15 =end
16
17 require 'csv'
18 student = Array.new
19 student1 = Array.new
20
21 Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
22   file_name = file_name1.split("/").last
23   first_name = file_name.split("/").last.split("_").first
24   p first_name
25   last_name = file_name.split("/").last.split("_",2).last.split("_").first
26   #task = file_name.split("_").last.split(".").first
27   student << ["#{first_name}", "#{last_name}"]
28 end
29
30 Dir.glob(ARGV[1]+"/**/*.*").each do |file_name1|
31   file_name = file_name1.split("/").last
32   first_name = file_name.split("/").last.split("_").first
33   p first_name
34   last_name = file_name.split("/").last.split("_",2).last.split("_").first
35   #task = file_name.split("_").last.split(".").first
36   student1 << ["#{first_name}", "#{last_name}"]
37 end
38
39 CSV.open("result.csv", "w") do |csv|
40   student.each do |fn, ln|
41     student1.each do |fn1, ln1|
42       if fn != fn1
43         if ln != ln1
44           csv << ["#{fn1}", "#{ln1}"]
45         end
46       end
47     end
48   end
49 end

```

The CSV is kind of odd. The code can be made better by using Array operators & and –

The program is almost ready but has to be fixed.

Mark: 3/5



```

1  #Develop a program named FirstName_LastName_ClassNumber_f70059.rb
2
3  #1. you are given two arguments for a folders with files;
4  #1.1 if there are other arguments they should be discarded;
5  #2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names excluding
6  extension. If there are duplicates the file must be written only once.;
7  #3. Calculate the length of their names (including extensions) divided by 2 rounded to the smalles number;
8  #4. Sort the result by File name ;
9  #5. Produce a result in CSV format named result.csv:
10
11      #           File1,3
12      #           File2,4
13      #           ...
14      #           FileN,3
15
16  require 'csv'
17  hash = Hash.new
18  count = 0
19  Dir.glob(ARGV[0] + "/*.rb") do |file|
20      first = file.split(/\/\//).last
21      puts first
22
23      #for (i = 0;i < first.length;i+=1)
24      size = first.length
25      i = 0
26      first.each do |element|
27
28          c = first[i].chr
29          if element == 0 || element == 1 || element == 2 || element == 3 || element == 4 || element == 5 ||
30             element == 6 || element == 7 || element == 8 || element == 9
31              count +=1
32          end
33      end
34      puts count
35  end
36
37  Dir.glob(ARGV[1] + "/*.rb") do |secFile|
38      sec = secFile.split(/\/\//).last
39      #puts sec
40
41  end
42
43  CSV.open("result.csv", "w") do |csv|
44      hash.sort_by{|key,val| key}.each do |element|
45          csv << element
46      end
47  end

```

He forgot to use the split("\_") method.

He has far too many mistakes.

Mark: 0/5

```

1 #Develop a program named FirstName_LastName_ClassNumber_627d43.r#
2 #
3 #1. you are given two arguments for a folders with files;
4 #1.1 if there are other arguments they should be discarded;
5 #2. file names in this folders are in the form First_Last_digits.rb;
6 #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;
7 #4. Sort the result by Last name ;
8 #5. Produce a result in CSV format named result.csv:
9 #
10 # LastName1,FirstName1
11 # LastName2,FirstName2
12 # ...
13 # LastNameN,FirstNameN
14
15
16
17 require 'csv'
18 name_array = Array.new()
19 name_array2 = Array.new()
20 support_array = Array.new()
21 support_array2 = Array.new()
22 i = 0
23 dir1 = ARGV[0]
24 dir2= ARGV[1]
25
26 Dir.glob("#{dir1}/*.rb") do |file|
27   name_array[i] = file.split(/\/\//).last
28   i += 1
29 end
30 count = i
31 i = 0
32 Dir.glob("#{dir2}/*.rb") do |file2|
33   name_array2[i] = file2.split(/\/\//).last
34   i += 1
35 end
36 i = 0
37 for check in i..count
38   if name_array[check] != name_array2[check]
39     support_array[i] = name_array[check]
40     support_array2[i] = name_array2[check]
41     i += 1
42     puts support_array
43     puts support_array2
44     CSV.open("result.csv", "w") do |csv|
45       support_array.each do |element|
46         csv << [element]
47       end
48     end
49     CSV.open("result.csv", "w") do |csv|
50       support_array2.each do |element2|
51         csv << [element2]
52       end
53     end
54   end
55 end

```

The problem is outputting too much and the actual filter mechanism doesn't work.

Mark: 2/5



```

1 #Develop a program named FirstName_LastName_ClassNumber_d77aee.rb
2 #
3 #1. you are given two arguments for a folders with files;
4 #1.1 if there are other arguments they should be discarded;
5 #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;
6 #3. Calculate the length of their names (including extensions).;
7 #4. Sort the result by length ;
8 #5. Produce a result in CSV format named result.csv:
9 #
10 #      File1,3
11 #      File2,4
12 #      ...
13 #      FileN,3
14
15 require 'csv'
16
17 first_folder = ARGV.shift
18 second_folder = ARGV.shift || "err"
19 names_hash = Hash.new
20
21 Dir.glob(first_folder+"/*.*").each do |text_file|
22   text_file = text_file.split("/").last
23   if (text_file.split("_").length == 3) then
24     first_name = text_file.split("_")[0]
25     second_name = text_file.split("_")[1]
26     diggit = text_file.split("_")[2].split(/\./).first
27     if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
28     if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
29     if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
30   else
31     names_hash[text_file] = text_file.length
32   end
33 end
34
35 if second_folder != "err"
36   Dir.glob(second_folder+"/*.*").each do |text_file|
37     text_file = text_file.split("/").last
38     if (text_file.split("_").length == 3) then
39       first_name = text_file.split("_")[0]
40       second_name = text_file.split("_")[1]
41       diggit = text_file.split("_")[2].split(/\./).first
42       if (diggit.to_i.to_s != diggit) then names_hash[text_file] = text_file.length end
43       if (first_name =~ /\d/) then names_hash[text_file] = text_file.length end
44       if (second_name =~ /\d/) then names_hash[text_file] = text_file.length end
45     else
46       names_hash[text_file] = text_file.length
47     end
48   end
49 end
50
51 names_hash = Hash[names_hash.sort_by{|k,v| k} ]
52 names_hash = Hash[names_hash.sort_by{|k,v| v} ]
53
54 puts names_hash
55
56 CSV.open("results.csv","w") do |csv|
57   names_hash.each do |element|
58     csv << element
59   end
60 end

```

Wrong output. The sorting is wrong. At least he sorted his hash before the CSV writing. Its hard to fix the code.

Mark: 3/5

```

1 =begin
2
3 Develop a program named FirstName_LastName_ClassNumber_041472.rb
4
5 1. you are given two arguments for a folders with files;
6 1.1 if there are other arguments they should be discarded;
7 2. file names in this folders are in the form First_Last_digits.rb;
8 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;
9 4. Sort the result by Last name ;
10 5. Produce a result in CSV format named result.csv:
11
12     LastName1,FirstName1
13     LastName2,FirstName2
14     ...
15     LastNameN,FirstNameN
16
17
18 =end
19
20 students_first_dir = Array.new
21 students_second_dir = Array.new
22
23 for i in 0..1
24
25     directory = ARGV[i]
26     if ARGV[i].split(//).last(1).to_s == "/"
27         directory += "**/*.rb"
28     else
29         directory += "/*.rb"
30     end
31
32     Dir.glob(directory).each do |dir|
33         student = dir.split(/\//)
34         if i == 0
35             students_first_dir.push(student)
36         else
37             students_second_dir.push(student)
38         end
39     end
40 end
41
42 studentcsv = Array.new
43
44 students_first_dir.each do |std|
45     match = 0
46     students_second_dir.each do |std2|
47         name = std.last.split(/_/)
48
49         name2 = std2.last.split(/_/)
50         for i in 0..1
51             if name[i] == name2[i]
52                 match = 1
53             end
54         end
55
56         end
57         studentcsv.push(name[1], name[2])
58     end
59
60 CSV.open("result.csv", "w") do |csv|
61     studentcsv.each do |string|
62         csv << string
63     end
64 end

```

Wrong. Name is not defined.

Mark: 1/5

```

1  #==begin
2  #Develop a program named FirstName_LastName_ClassNumber_bce70c.rb
3  #
4  #1. you are given an argument for a folder with files;
5  #1.1 if there are other arguments they should be discarded
6  #2. file names in this folder are in the form First_Last_digits.rb;
7  #3. find all the students that have 5 letters in their second name;
8  #4. Sort the result by First name DESC.
9  #5. Produce a result in CSV format named result.csv:
10 #
11 #      FirstName1,LastName1
12 #      FirstName2,LastName2
13 #      ...
14 #      FirstNameN,LastNameN
15 #==end
16
17 require 'csv'
18
19 hash = Hash.new
20
21 Dir.glob("#{ ARGV[0] }/*") do |name|
22   name = name.split("/").last
23   short_name = name.split('_')[1]
24   if short_name.length == 5
25     hash[name] = short_name
26   end
27 end
28
29 CSV.open("result.csv", "w") do |csv|
30   hash = hash.sort_by { |key, value| value }.reverse
31   hash.each |key| do
32     csv << key
33   end
34 end

```

Problem: ☐ Wrong file name and wrong each loop.

Fix: ☐ Change file name to: Kalin\_Marinov\_12\_bce70c.rb

And change “hash.each |key| do” to “hash.each do |key|”

Mark: 3/5

```

1  =begin
2  Develop a program named FirstName_LastName_ClassNumber_0af18f.rb
3
4  1. you are given an argument for a folder with files;
5  1.1 if there are other arguments they should be discarded
6  2. file names in this folder are in the form First_Last_digits.rb;
7  3. find all the students that have 5 letters in their second name;
8  4. Sort the result by First name DESC.
9  5. Produce a result in CSV format named result.csv:
10
11      FirstName1,LastName1
12      FirstName2,LastName2
13      ...
14      FirstNameN,LastNameN
15  =end
16
17  student = Hash.new { |name, programs| name[programs] = []}
18  directory = ARGV[0]
19  require "csv"
20
21  class String
22      def is_number?
23          Float(self) != nil rescue false
24      end
25  end
26
27  Dir.glob("#{directory}/*.rb") do |my_repository|
28
29      name_dir = my_repository.split("/").last
30
31      name = name_dir.split("_").first.capitalize
32      sir_name = name_dir.split("_", 2).last.split("_").first.capitalize
33      program = name_dir.split("_").last.split(".").first
34      ex = name_dir.split("_").last.split(".").last
35
36      if name_dir.include? "_" then counter = name_dir.count "_" end
37      student["#{name}"] << sir_name if ((counter == 2) && (sir_name.length == 5) && (program.is_number?) && (ex ==
38          "rb"))
39  end
40  CSV.open("result.csv", "w") do |csv|
41      student.sort_by{|k, v| v}.reverse.each do |f_name, l_name|
42          csv << [f_name, l_name].flatten
43      end
44  end

```

Problem: Wrong sorting.

Fix: Change “student.sort\_by{|k, v| v}.reverse.each” to “student.sort\_by{|k, v| k}.reverse.each”

Mark: 4/5



```

1  =begin
2  Develop a program named FirstName_LastName_ClassNumber_890ba0.rb
3
4  1. you are given an argument for a folder with files;
5  1.1 if there are other arguments they should be discarded
6  2. file names in this folder are in the form First_Last_digits.rb;
7  3. find all the students that have 10 letters in their first name;
8  4. Sort the result by Last Name DESC.
9  5. Produce a result in CSV format named result.csv:
10
11      FirstName1,LastName1
12      FirstName2,LastName2
13      ...
14      FirstNameN,LastNameN
15  =end
16
17
18  require 'csv'
19  results=Hash.new
20  Directory = ARGV[0]
21  Dir.glob("#{Directory}/*.rb") do |file_name|
22
23      first_name = file_name.split("/").last.split("_").first.capitalize
24      last_name=file_name.split("/").last.split("_",2).last.split("_").first.capitalize
25
26      if first_name.length == 10
27
28          results["#{last_name}"] = "#{first_name}"
29      end
30  end
31
32
33
34  CSV.open("results.csv", "w") do |csv|
35      results.sort.each do |first,last|
36
37          csv << [last,first]
38      end
39  end
40 end

```

Problem: ☐ Wrong result file name and sorting.

Fix: Change: "results.csv" to "result.csv"

Change "result.sort.each" to "result.sort.reverse.each"

Mark: 4/5

```

1  require 'csv'
2  def is_numeric(o)
3      true if Integer(o) rescue false
4  end
5
6  array = []
7  count = 0
8
9  Dir.glob(ARGV[0] + "*").each do |file|
10     ch_count = 0
11     file_name = file.split("/").last.split("")
12
13     file_name.each do |ch|
14
15         if is_numeric(ch)
16
17             ch_count += 1
18
19         end
20
21     end
22     if ch_count == 9
23         len = file_name.length
24         array[count] = []
25         array[count][0] = file_name
26         array[count][1] = len/2.round
27         count += 1
28     end
29
30 end
31 array = array.sort_by {|el| el[0]}
32 CSV.open("results.csv", "w") do |csv|
33
34     array.each do |element|
35
36         csv << element
37
38     end
39
40 end

```

Problem: A lot of errors.

Mark: 1/5

```

1  =begin
2  Develop a program named FirstName_LastName_ClassNumber_b7f153.rb
3
4  1. you are given an argument for a folder with files;
5  1.1 if there are other arguments they should be discarded
6  2. file names in this folder are in the form First_Last_digits.rb;
7  3. find all the students that have 10 letters in their second name;
8  4. Sort the result by Last Name ASC.
9  5. Produce a result in CSV format named result.csv:
10
11      FirstName1,LastName1
12      FirstName2,LastName2
13      ...
14      FirstNameN,LastNameN
15  =end
16  require 'csv'
17
18  student = Hash.new
19
20  Dir.glob("#{ARGV[0]}*_*_.rb") do |file|
21      firstName = file.split("/").last.split("_").first
22      lastName = file.split("/").last.split("_", 2).last.split("_").first
23      digit = file.split("/").last.split("_").last.split(".").first
24      if lastName.length == 10
25          student[firstName] = lastName
26      end
27  end
28
29  CSV.open("result.csv", "w") do |csv_file|
30      student.sort.each do |key, value|
31          csv_file << ["#{key}, #{value}"]
32      end
33  end

```

Change “csv\_file << [“#{key}, #{value}”]” to “csv\_file << [“#{key}, #{value}”]”

Mark: 4/5