Презентация за грешките и възможните решения от контролното

Technical School Electronic Systems

October.2014

Team:
RubyDevelopers

From: Hristiyan Velyakov, Veselin Dechev

Report

Common mistakes

1. The most common mistake is having the output CSV file wrong named.
2. Another mistake is undefined method in few programs
3. Another frequent mistake is the wrong sorting of the output data
4. There are have few programs with syntax errors

Common solutions:

1. In many of the programs the errors are caused by misreading, or misunderstanding. So the most common solution is paying enough **attention**
2. 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.

Borislav Rusinov

has program which works correctly .The problem is only csv name.

Produce a result in CSV format named result.csv:
File.open("results.csv", "w") do |csv|

Denis Trenchev

has program with many wrongs .Few of them are incorrect csv name, undefined array .The program check second name for 5 letters instead first name.

Produce a result in CSV format named result.csv:
File.open("students.csv", "w") do |csv|
find the students with 5 letters in the first name
if name[1].to_s.length == 5

Dimitar Nestorov

has program which has 2 problems .First of them is program check second name for 10 letters instead first name.Second problems is error by sort.

find all the students that have 10 letters in their first name;
if name[1].length == 10
array = array.sort_by {|el| -el[1]}-error

Kristina Pironkova

has program which works correctly .The problem is only csv name.

Produce a result in CSV format named result.csv:
File.open("results.csv", "w") do |csv|

Lubomir Yankov

his program check for 9 digits in name instead 7 and wrong csv name .

if ch_count == 9 must be if ch_count == 7

Petko Bozhinov

his program works but in some cases doesn't work.It doesn't work when we have wrong name like this Petko_Bozhinov.rb We can solve this problem . If we put one "if" which check for name size.

Stanislav Gospodinov

has program which works correctly .The problem is only csv name.

Produce a result in CSV format named result.csv:
File.open("results.csv", "w") do |csv|

Veselin Dechev

his problem has few problems.1-st of them is split string wrong .It takes name including extensions.Other problem is program compare names from both folders wrong.It compare for equal names instead output names which are only on 1-st folder.

Tihomir Lidanski

has program which has a little part of code .There missing big parts from code .

Stanislav Vulkanov

his program has little problem.With this row
If(last.length==5)&&(short_name.split("_").size==3) if we reverse this will work correctly

Radoslav Kostadinov

has problem is program compare names from both folders wrong.It compare for equal names instead output names which are only on 2-nd folder.

Dimitar Terziev

His program works properly ,but some parts of code are hard to read

Radoslav Kostadinov

His program search for equal names in both folders instead output unique from 2-nd folder.

if first_name1 == first_name and last_name1 == last_name-wrong

Simeon Shopkin

His program has undefined method for array

David Georgiev

```
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]}"])
```

Wrong split

Не сортира Last Name по ASC >> Sort the result by Last Name ASC.

Borislav Stratev

```
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 - грешно
full_first_name = full_first_name.to_s
tmp = file_name.split("#{first_name}_")
full_last_name = tmp.last.split(/_/).first.to_s + digit - грешно
full_last_name = full_last_name.to_s
h[full_last_name] = full_first_name                >>>
```

LastName + digits , FirstName + digits to: LastName1 , FirstName1

- Find the students that are only in the first folder and not in the second. A student is in both folders if there is a file with the same First and Last Name. Digits might be different

results.csv to result.csv

Lili Kokalova

```
Dir.glob(ARGV[0]+"/**/*.*").each do |file_name1|
..
..
Dir.glob(ARGV[1]+"/**/*.*").each do |file_name1|
```

file_name1 is already taken

Don’t output in format

LastName1,FirstName1

LastName2,FirstName2

...

LastNameN,FirstNameN

csv << ["#{fn1}", "#{ln1}"] Should be in<< ["#{ln1}", "#{fn1}"]

Iliyan Germanov

Don’t sort properly
4. Sort the result by Last name ;

Nikolay Mihailov

#2. Find all the files from both folders that have exactly 7 digits from 0 to 9 in their names excluding extension.
Don’t split properly>> first = file.split(/\//).last

Не го ограничава до само 7 цифри >> that have exactly 7 digits

```
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
```

#3. Calculate the length of their names (including extensions) divided by 2 rounded to the smalles number;

This is missing in the code

Valentin Varbanov

first folder and not in the second

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

Не е в този формат

Stefan Iliev

Wrong split>> that are not in the format FirsrName_LastName_digit.rb

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

Stanislav Iliev

Don’t split in format>> form First_Last_digits.rb
Don’t sort properly>> #4. Sort the result by Last name ;

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

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

```
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]
```

Write in twice CSV

Veselina Kolova

if File.extname(text_file) text_file.include?(".rb") && text_file.split(/_/).last.split(/\./).first.to_i.is_a Integer then   << error in include?
>>solution >> Dir.glob(ARGV[0] + "*.rb").each do |text_file| >> now bold text is not neccesery
Wrong sort>> LastName1,FirstName1

Vladimir Yordanov

Wrong Sort #4. Sort the result by Last Name ASC.

#5. Produce a result in CSV format named result.csv: > > results.csv to result.csv