

このページの内容を全て印刷し、提出してください。

チェック時刻：2018/12/14 14:49:26_703

チェック結果

自動チェックによるコメント不足は見当りませんでした。(コメントについて問題無いという意味ではありません。)

自動チェックによる行の長さの問題は見当りませんでした。

注意：チェックシステムの問題により、プログラム内で使われているシャープ記号(#)が、誤ってコメントの記号と扱われ「コメントが短い」と判定される場合があります。システムがこのような指摘をした場合、無視して、問題無し、と判断してください。

ソースプログラム

```
1  #standard inputs to file "posts-out.txt"
2  $stdout = File.open('posts-out.txt', 'w')
3  require 'time'
4  require 'csv'
5  require "fileutils"
6  #erase merged contain
7  File.open('out.csv','w'){|file| file = nil}
8  #search for csv files and load it
9  Dir.glob('posts*.csv').each {|csvpath|
10   # "posts~.csv" files merge "out.csv"
11   csvdata = CSV.read(csvpath)
12   CSV.open('out.csv','a') {|merge|
13     for data in csvdata
14       merge.puts data
15     end
16   }
17 }
18 #head of table(this is the templete of result-data)
19 print("user_id,")
20 for hours in 0..22
21   print(hours, ",")
22 end
23 print("23\n")
24 #load a csv file
25 File.open('out.csv'){|csvdata|
26   #detect num of users
27   usernum = csvdata.read.scan(/user[0-9]+/).uniq.size
28   #make new 2D-array to save the result of analysis
29   rec = Array.new(usernum).map{Array.new(24,0)}
30   #load each line of a csv file
31   File.readlines(csvdata).each {|csveach|
32     #split into strings in an array
33     csveachs = csveach.split(",")
34     #make string 0:00:00~23:59:59 treat as time class and change into array
```

```
35     csveachs[1] = Time.strptime(csveachs[1], "%H:%M:%S").to_a
36     #change "userXXXXXXXX" into "X(1-degit)"
37     csveachs[2].gsub!(/user0+/, "")
38     #detect post-time and save this count
39     rec[csveachs[2].to_i-1][csveachs[1][2]] += 1
40 }
41 rec = rec.map{|a|a.map{|b|b = b **2}}
42 #shape appearance of result-data following example templete
43 for usernum_e in 1..usernum
44     #change "X(1-degit)" into "userXXXXXXXX"
45     usernum_8 = '%08d' % usernum_e
46     anldata = "user" + usernum_8 + "," + rec[usernum_e-1].join(",")
47     puts anldata
48 end
49 }
50 FileUtils.rm('out.csv')
```