

Zadatak

U datoteci koja se zadaje kao argument naredbenog retka (ne mora biti smještena u tekućem direktoriju) pohranjene su e-mail poruke razmjenjivane preko jedne distribucijske liste. Svaka poruka započinje praznim retkom, nakon čega dolazi redak koji započinje nizom "From nobody". U tom su retku zapisani i podaci o datumu i vremenu primitka elektroničke poruke. Primjer ovakve datoteke je: 2017-04.mail

Svaka poruka sadrži redak u zaglavlju koji započinje nizom "From: ", nakon čega dolazi ime osobe koja je slala poruku te adresa elektroničke pošte te osobe, navedena unutar znakova "<" i ">".

Napisati perl skriptu koja će ispisati histogram slanja pojedine osobe, tj. svaki redak ispisa sastoji se od elektroničke adrese osobe poravnate u desno, potom dolazi niz zvjezdica (*) čiji broj odgovara broju poruka koje je ta osoba poslala, te redak završava ispisanim brojem poslanih poruka u zagradi. Ispis treba biti poredan po rastućem broju poslanih poruka, pri čemu poredak pošiljatelja s istim brojem poruka nije bitan. Pripaziti da se pronalaze isključivo ispravne adrese elektroničke pošte (oblika identifikator@domena).

Primjer izvršavanja skripte zadatak2.pl:

```
$ perl zadatak2.pl ../TestPrimjeri/2017-04.mail

      morrowc.lists@gmail.com: * (1)
      kolkman@isoc.org: * (1)
      vinayakh@gmail.com: * (1)
      ...
      mirja.kuehlewind@tik.ee.ethz.ch: **** (4)
      Jonathan.Hardwick@metaswitch.com: **** (4)
      job@instituut.net: **** (4)
      sabine.randriamasy@nokia-bell-labs.com: ****
      (4)
      nishida@sfc.wide.ad.jp: ***** (5)
      john-ietf@jck.com: ***** (5)
      mstjohns@comcast.net: ***** (5)
      johnl@taugh.com: ***** (5)
      ...
      stewart.bryant@gmail.com: ***** (10)
      phill@hallambaker.com: ***** (11)
      martin.thomson@gmail.com: ***** (11)
      nico@cryptonector.com: ***** (12)
      ...
```

Zadatak

Napisati perl skriptu koja će učitati zadanu datoteku s popisom laboratorijskih termina dobivenim iz FERweba ispisati podatke o laboratorijskim terminima. Ulazna datoteka u svakom retku sadrži podatke jednoj laboratorijskoj grupi, a polja su odvojena tabulatorima ("\t"). Primjer datoteke je [labosi.txt](#). Polja su: naziv grupe, opis vježbe, broj studenata u grupi, datum i dan održavanja, vrijeme održavanja, trajanje u minutama, te prostorija u kojoj se vježba održava. Ispis treba sortirati kronološki prema terminima i oblikovati prema primjeru u nastavku. Ispisuje se termin, vrijeme održavanja i dvorana.

Poziv skripte zadatak1.pl:

```
$ perl zadatak1.pl labosi.txt
13.05.2013. (pon)      08:00 - 11:00  A210
13.05.2013. (pon)      10:00 - 13:00  A209
13.05.2013. (pon)      11:00 - 14:00  A210
14.05.2013. (uto)      08:00 - 11:00  A209
14.05.2013. (uto)      11:00 - 14:00  A210
14.05.2013. (uto)      17:00 - 20:00  A209
15.05.2013. (sri)      10:00 - 13:00  A209
15.05.2013. (sri)      17:00 - 20:00  A209
16.05.2013. (čet)      08:00 - 11:00  A209
16.05.2013. (čet)      14:00 - 17:00  A209
16.05.2013. (čet)      17:00 - 20:00  A210
17.05.2013. (pet)      08:00 - 11:00  A209
17.05.2013. (pet)      13:00 - 16:00  A209
```

```
@termini = ();

while(<>) {
    push @termini, $_;
}

@termini = sort { (split /\t/, $a)[3] <=> (split /\t/, $b)[3] || (split /\t/, $a)[4] <=> (split /\t/, $b)[4]} @termini;

#ispis
for ($i = 0; $i <= $#termini; $i++) {
    print ((split /\t/, $termini[$i])[3], "\t", (split /\t/, $termini[$i])[4], "\t", (split /\t/, $termini[$i])[6]);
    print "\n";
}

ILI

#!/usr/bin/perl -w

if (@ARGV != 1){
    die "die";
}
while (<>) {
    @row = split /\t/, $_;
    $hash{$row[3]."\t".$row[4]} = $row[6];
}
foreach $key (sort keys %hash){
    print($key."\t".$hash{$key}."\n");
}
```

Zadatak

U log datoteci nalazi se zapisi o (ne)uspješnim pokretanjima aplikacija i servisa. Svaki zapis log datoteke pripada jednoj od tri razine obavijesti: pogreška (Error), upozorenje (Warning) ili informacija (Information). Za svaki zapis bilježe se sljedeći podaci: razina (Level), datum i vrijeme (Date and Time), izvor (Source), ID događaja (EventID), zadatak (Task) i kategorija (Category). Podaci pojedinog zapisa međusobno su razdvojeni tabom ("\t"). Primjer log datoteke je application_event_log.txt.

Napisati perl skriptu koja će učitati log datoteku koja se zadaje kao argument naredbenog retka, te za svaku od tri razine obavijesti (Information, Error, Warning) odrediti koliko je puta pojedini izvor (Source) generirao obavijest te razine. Ispis sortirati po nazivu izvora i oblikovati prema primjeru u nastavku.

Poziv skripte zadatak4.pl:

```
$ perl zadatak4.pl application_event_log.txt
```

```
=====
Level: Error
Source: num. records
-----
Chrome Report Engine : 1
Microsoft-Windows-Defrag : 1
Report Server Windows Service (SQLR2) : 5
Windows Error Reporting : 3

=====
Level: Information
Source: num. records
-----
Chrome Report Engine : 4
Internet Explorer 8 Startup Reporting : 1
Windows Error Reporting : 10

=====
Level: Warning
Source: num. records
-----
Chrome Report Engine : 1
Internet Explorer 8 Startup Reporting : 3
Windows Error Reporting : 1
```

DATOTEKA

Level	Date and Time	Source	Event ID	Task	Category
Error	9.5.2013. 10:29:00	Report Server Windows Service (SQLR2)	107	Management	Report
Server Windows Service (SQLR2) cannot connect to the report server database.					
Error	9.5.2013. 10:28:57	Report Server Windows Service (SQLR2)	107	Management	Report
Server Windows Service (SQLR2) cannot connect to the report server database.					

Error	9.5.2013. 9:08:36	Microsoft-Windows-Defrag	258	None	The disk defragmenter successfully completed defragmentation on \\?\Volume{64c4f355-a06a-11e1-86e7-806e6f6e6963}\	
Information	10.5.2013. 8:25:21	Windows Error Reporting	1001	None	"Fault bucket 1946324623, type 5"	
Warning	9.5.2013. 8:25:04	Internet Explorer 8 Startup Reporting	1001	None	"Fault bucket , type 0"	
Information	9.5.2013. 8:25:04	Chrome Report Engine	1001	None	"Fault bucket , type 0"	
Error	10.5.2013. 10:29:00	Report Server Windows Service (SQLR2)	107	Management	Report Server Windows Service (SQLR2) cannot connect to the report server database.	
Information	9.5.2013. 8:15:13	Windows Error Reporting	1001	None	"Fault bucket 3433144111, type 5"	
Information	9.5.2013. 8:15:12	Internet Explorer 8 Startup Reporting	1001	None	"Fault bucket 3568120217, type 5"	
Error	10.5.2013. 10:29:00	Report Server Windows Service (SQLR2)	107	Management	Report Server Windows Service (SQLR2) cannot connect to the report server database.	
Information	9.5.2013. 8:15:11	Windows Error Reporting	1001	None	"Fault bucket 3498954754, type 5"	
Error	9.5.2013. 8:15:11	Windows Error Reporting	1001	None	"Fault bucket 3478351210, type 5"	
Information	9.5.2013. 8:15:11	Chrome Report Engine	1001	None	"Fault bucket 3370976229, type 5"	
Error	9.5.2013. 10:29:00	Report Server Windows Service (SQLR2)	107	Management	Report Server Windows Service (SQLR2) cannot connect to the report server database.	
Information	9.5.2013. 8:15:11	Windows Error Reporting	1001	None	"Fault bucket 3568119963, type 5"	
Information	9.5.2013. 8:15:11	Windows Error Reporting	1001	None	"Fault bucket 3433143983, type 5"	
Warning	9.5.2013. 8:15:11	Internet Explorer 8 Startup Reporting	1001	None	"Fault bucket 3370976231, type 5"	
Information	9.5.2013. 8:15:11	Windows Error Reporting	1001	None	"Fault bucket 3290924375, type 5"	
Warning	9.5.2013. 8:15:10	Internet Explorer 8 Startup Reporting	1001	None	"Fault bucket 3519140734, type 5"	
Information	9.5.2013. 8:15:10	Chrome Report Engine	1001	None	"Fault bucket 3478330384, type 5"	
Error	9.5.2013. 8:15:10	Windows Error Reporting	1001	None	"Fault bucket 3478330391, type 5"	
Information	9.5.2013. 8:15:10	Windows Error Reporting	1001	None	"Fault bucket 3593420390, type 5"	

Information	9.5.2013. 9:15:10	Windows Error Reporting	1001	None	"Fault bucket 3433161327, type 5"
Warning	9.5.2013. 10:17:10	Chrome Report Engine	1001	None	"Fault bucket 3433149329, type 5"
Information	9.5.2013. 8:15:10	Windows Error Reporting	1001	None	"Fault bucket 3370982485, type 5"
Error	9.5.2013. 8:15:10	Windows Error Reporting	1001	None	"Fault bucket 3370982637, type 5"
Warning	9.5.2013. 8:15:09	Windows Error Reporting	1001	None	"Fault bucket 3370980240, type 5"
Information	9.5.2013. 8:15:09	Windows Error Reporting	1001	None	"Fault bucket 3519161488, type 5"
Error	9.5.2013. 8:15:09	Chrome Report Engine	1001	None	"Fault bucket 3401091840, type 5"
Information	9.5.2013. 8:15:09	Chrome Report Engine	1001	None	"Fault bucket 3327313096, type 5"

RJESNJE

```
open FILE, $ARGV[ $#ARGV];
```

```
chomp(@lines = <FILE>); shift @lines;
```

```
for (@lines) {
```

```
    ($level, $dt, $source, $event, $task, $category) = split(/\t/, $_);
    $map{$level}{$source}++;
```

```
}
```

```
foreach $level (sort keys %map) {
```

```
    print "===== =\n";
    print " Level: $level\n";
    print " Source: num.records\n";
    print "-----\n";
```

```
    foreach $source (sort keys %map{$level}) {
        print "$source : $map{$level}{$source}\n";
    }
    print "\n";
```

```
}
```

Napisati perl skriptu koja će učitati zadanu datoteku s popisom laboratorijskih termina dobivenim iz FERweba ispisati podatke o laboratorijskim terminima. Ulazna datoteka u svakom retku sadrži podatke jednoj laboratorijskoj grupi, a polja su odvojena tabulatorima ("t"). Primjer datoteke je labosi.txt. Polja su: naziv grupe, opis vježbe, broj studenata u grupi, datum i dan održavanja, vrijeme održavanja, trajanje u minutama, te prostorija u kojoj se vježba održava. Ispis treba sortirati kronološki prema terminima i oblikovati prema primjeru u nastavku. Ispisuje se redni broj termina, datum, vrijeme održavanja, dvorana i broj studenata u grupi.

Poziv skripte zadatak1.pl:

```
$ perl zadatak1.pl labosi.txt
```

```
1. 13.05.2013. (pon) 08:00 - 11:00 A210 15
2. 13.05.2013. (pon) 10:00 - 13:00 A209 16
3. 13.05.2013. (pon) 11:00 - 14:00 A210 15
4. 14.05.2013. (uto) 08:00 - 11:00 A209 16
5. 14.05.2013. (uto) 11:00 - 14:00 A210 16
6. 14.05.2013. (uto) 17:00 - 20:00 A209 16
7. 15.05.2013. (sri) 10:00 - 13:00 A209 14
8. 15.05.2013. (sri) 17:00 - 20:00 A209 16
9. 16.05.2013. (čet) 08:00 - 11:00 A209 15
10. 16.05.2013. (čet) 14:00 - 17:00 A209 15
11. 16.05.2013. (čet) 17:00 - 20:00 A210 16
12. 17.05.2013. (pet) 08:00 - 11:00 A209 15
13. 17.05.2013. (pet) 13:00 - 16:00 A209 15
```

DATOTEKA

LAB02-01	Skriptni jezici - 2. laboratorijska vježba (LAB02-01)	16	14.05.2013. (uto)	08:00 - 11:00	180	A209
LAB02-02	Skriptni jezici - 2. laboratorijska vježba (LAB02-02)	16	14.05.2013. (uto)	17:00 - 20:00	180	A209
LAB02-03	Skriptni jezici - 2. laboratorijska vježba (LAB02-03)	15	16.05.2013. (čet)	14:00 - 17:00	180	A209
LAB02-04	Skriptni jezici - 2. laboratorijska vježba (LAB02-04)	16	15.05.2013. (sri)	17:00 - 20:00	180	A209
LAB02-05	Skriptni jezici - 2. laboratorijska vježba (LAB02-05)	15	13.05.2013. (pon)	08:00 - 11:00	180	A210
LAB02-06	Skriptni jezici - 2. laboratorijska vježba (LAB02-06)	16	16.05.2013. (čet)	17:00 - 20:00	180	A210
LAB02-07	Skriptni jezici - 2. laboratorijska vježba (LAB02-07)	15	17.05.2013. (pet)	08:00 - 11:00	180	A209
LAB02-08	Skriptni jezici - 2. laboratorijska vježba (LAB02-08)	16	14.05.2013. (uto)	11:00 - 14:00	180	A210
LAB02-09	Skriptni jezici - 2. laboratorijska vježba (LAB02-09)	14	15.05.2013. (sri)	10:00 - 13:00	180	A209
LAB02-10	Skriptni jezici - 2. laboratorijska vježba (LAB02-10)	15	13.05.2013. (pon)	11:00 - 14:00	180	A210

LAB02-11	Skriptni jezici - 2. laboratorijska vježba (LAB02-11)	15	16.05.2013. (čet)	08:00
- 11:00	180	A209		

LAB02-12	Skriptni jezici - 2. laboratorijska vježba (LAB02-12)	16	13.05.2013. (pon)	10:00
- 13:00	180	A209		

LAB02-13	Skriptni jezici - 2. laboratorijska vježba (LAB02-13)	15	17.05.2013. (pet)	13:00
- 16:00	180	A209		

```
#!/perl
```

```
use open ':locale';
```

```
use locale;
```

```
#prima jednu datoteku
```

```
while(<>){
```

```
@linija = split "\t", $_;
```

```
push @popis, "$linija[3]\t$linija[4]\t$linija[6]\t$linija[2]";
```

```
}
```

```
foreach (sort @popis){
```

```
print ++$cnt . ".\t$_\n";
```

```
}
```

U log datoteci nalazi se zapisi o (ne)uspješnim pokretanjima aplikacija i servisa. Svaki zapis log datoteke pripada jednoj od tri razine obavijesti: pogreška (Error), upozorenje (Warning) ili informacija (Information). Za svaki zapis bilježe se sljedeći podaci: razina (Level), datum i vrijeme (Date and Time), izvor (Source), ID događaja (EventID), zadatak (Task) i kategorija (Category). Podaci pojedinog zapisa međusobno su razdvojeni tabom ("\t"). Primjer log datoteke je application_event_log.txt.

Napisati perl skriptu koja će učitati log datoteku koja se zadaje kao argument naredbenog retka, te za svaku od tri razine obavijesti (Information, Error, Warning) odrediti koliko je puta pojedini izvor (Source) generirao obavijest te razine. Ispis sortirati silazno prema broju generiranih obavijesti i oblikovati prema primjeru u nastavku.

Poziv skripte zadatak2.pl:

```
$ perl zadatak2.pl application_event_log.txt
```

```
=====
Level: Error
Source: num. records
-----
Report Server Windows Service (SQLR2) : 5
Windows Error Reporting : 3
Chrome Report Engine : 1
Microsoft-Windows-Defrag : 1

=====
Level: Information
Source: num. records
-----
Windows Error Reporting : 10
Chrome Report Engine : 4
Internet Explorer 8 Startup Reporting : 1

=====
Level: Warning
Source: num. records
-----
Internet Explorer 8 Startup Reporting : 3
Chrome Report Engine : 1
Windows Error Reporting : 1

%error;
%warning;
%information;
while(defined($ulaz=<>)){
    @redak = split /\t/, $ulaz;
    if(@redak[0] eq "Error"){
        $error{@redak[2]}++;
    }
    if(@redak[0] eq "Information"){
        $information{@redak[2]}++;
    }
}
```



```

}
if(@redak[0] eq "Warning"){
$warning{@redak[2]}++;
}
}
print "===== \n";
print " Level: Error\n";
print " Source: num. records\n";
print "-----\n";
foreach (sort {$error{$b} <=> $error{$a}} (keys %error)){
print $_. " : ".$error{$_}."\n";
}
print "\n";
print "===== \n";
print " Level: Information\n";
print " Source: num. records\n";
print "-----\n";
foreach (sort {$information{$b} <=> $information{$a}} (keys %information)){
print $_. " : ".$information{$_}."\n";
}
print "\n";
print "===== \n";
print " Level: Warning\n";
print " Source: num. records\n";
print "-----\n";
foreach (sort {$warning{$b} <=> $warning{$a}} (keys %warning)){
print $_. " : ".$warning{$_}."\n";
}
print "\n";
}
}

```

```

#!/usr/bin/perl

if(@ARGV < 1) {
    print "Invalid call\n";
    print "zadatak2.pl <log_file>\n";
    exit -1;
}
while(<>) {
    chomp;
    next if (/^Level/); # skip first line with info

    @parts = split /\t/;
    $lvl = $parts[0];
    $src = $parts[2];
    $cntr{$lvl}{$src}++;
}

foreach $type (keys %cntr) {
    #print header
    print "=" x 40 . "\n";
    printf " Level : %s\n", $type;
    print " Source : num. records\n";
}

```

```
print "-" x 40 . "\n";

@keys = sort { $cntr{$type}{$b} <=> $cntr{$type}{$a} } keys $cntr{$type};

foreach $src (@keys) {
    printf "%s : %s\n", $src, $cntr{$type}{$src};
}
print "\n";
}
```

ZADATAK

U log datoteci web poslužitelja zapisani su podaci o zahtjevima klijentskih računala. Podaci su navedeni sljedećim redom: IP adresa klijenta, dva polja s identifikatorima korisnika (u primjeru su ta polja prazna, odnosno označena znakom minus), polje s datumom i vremenom obrade zahtjeva (u uglatim zagradama), podaci o zahtjevu korisnika (unutar navodnika), oznaka statusa, i konačno veličina objekta vraćenog klijentu, izražena u bajtovima. Minus u pojedinom polju označava da podatak nije raspoloživ. Primjer log datoteke je `localhost_access_log.txt`.

Napisati perl skriptu koja će učitati log datoteku koja se zadaje kao argument naredbenog retka, te za svakog klijenta izbrojati pristupe poslužitelju. Ispis sortirati silazno prema broju pristupa i oblikovati prema primjeru u nastavku.

DATOTEKA

```
89.172.5.200 - - [24/Feb/2008:00:00:21 +0100] "GET /burza/b/Logout.action?ciid=26 HTTP/1.1" 302 -
89.172.5.200 - - [24/Feb/2008:00:00:21 +0100] "GET /burza/b/Main.action HTTP/1.1" 200 1111
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET / HTTP/1.1" 200 15118
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /images/java-color100.png HTTP/1.1" 200 11698
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /images/tomcat-1.png HTTP/1.1" 200 5641
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /favicon.ico HTTP/1.1" 404 988
78.3.194.174 - - [24/Feb/2008:00:04:34 +0100] "HEAD /zadaci.pdf HTTP/1.1" 200 -
78.3.194.174 - - [24/Feb/2008:00:04:35 +0100] "GET /zadaci.pdf HTTP/1.1" 200 86872
78.3.194.174 - - [24/Feb/2008:00:04:39 +0100] "GET /about.jsp?lang=hr HTTP/1.1" 200 4802
78.3.194.174 - - [24/Feb/2008:00:04:42 +0100] "GET /members.jsp?lang=hr HTTP/1.1" 200 7260
78.3.194.174 - - [24/Feb/2008:00:04:48 +0100] "GET /challenges.jsp?lang=hr HTTP/1.1" 200 10561
78.3.194.174 - - [24/Feb/2008:00:04:51 +0100] "GET /coursesTutorials.jsp?lang=hr HTTP/1.1" 200 6656
78.3.194.174 - - [24/Feb/2008:00:05:03 +0100] "GET /coursesTutorials.jsp?lang=hr HTTP/1.1" 200 6656
78.3.194.174 - - [24/Feb/2008:00:05:04 +0100] "GET /joinIn.jsp?lang=hr HTTP/1.1" 200 5034
78.3.194.174 - - [24/Feb/2008:00:05:24 +0100] "GET /about.jsp?lang=hr HTTP/1.1" 200 4802
78.0.95.36 - - [24/Feb/2008:00:10:19 +0100] "GET /burza/b/Main.action HTTP/1.1" 200 1111
78.0.95.36 - - [24/Feb/2008:00:10:19 +0100] "GET /burza/css/default.css HTTP/1.1" 200 2516
89.201.135.32 - - [24/Feb/2008:00:10:29 +0100] "GET /burza/b/Main.action?ciid=3 HTTP/1.1" 200 1425
89.201.135.32 - - [24/Feb/2008:00:10:29 +0100] "GET /burza/css/default.css?ciid=3 HTTP/1.1" 200 2516
89.201.135.32 - - [24/Feb/2008:00:10:32 +0100] "GET /burza/b/Login.action?ciid=3 HTTP/1.1" 200 1524
89.201.135.32 - - [24/Feb/2008:00:10:32 +0100] "GET /burza/css/default.css?ciid=3 HTTP/1.1" 200 2516
89.201.135.32 - - [24/Feb/2008:00:10:35 +0100] "POST /burza/b/Login.action HTTP/1.1" 302 -
89.201.135.32 - - [24/Feb/2008:00:10:35 +0100] "GET /burza/b/Main.action HTTP/1.1" 200 2093
89.201.135.32 - - [24/Feb/2008:00:10:35 +0100] "GET /burza/css/default.css HTTP/1.1" 200 2516
```

89.201.135.32 - - [24/Feb/2008:00:10:38 +0100] "GET /burza/b/BrowseCIMarketplace.action?ciid=1 HTTP/1.1" 200 2703

89.201.135.32 - - [24/Feb/2008:00:10:38 +0100] "GET /burza/css/default.css?ciid=1 HTTP/1.1" 200 2516

78.0.95.36 - - [24/Feb/2008:00:11:42 +0100] "GET /burza/b/Login.action HTTP/1.1" 200 1510

78.0.95.36 - - [24/Feb/2008:00:11:47 +0100] "POST /burza/b/Login.action HTTP/1.1" 302 -

78.0.95.36 - - [24/Feb/2008:00:11:47 +0100] "GET /burza/b/Main.action HTTP/1.1" 200 1598

78.0.95.36 - - [24/Feb/2008:00:12:55 +0100] "GET /burza/b/BrowseCIMarketplace.action?ciid=13 HTTP/1.1" 200 3465

78.0.95.36 - - [24/Feb/2008:00:12:57 +0100] "GET /burza/css/default.css?ciid=13 HTTP/1.1" 200 2516

78.0.95.36 - - [24/Feb/2008:00:12:57 +0100] "GET /burza/b/BrowseCIMarketplace.action?ciid=13 HTTP/1.1" 200 3465

78.0.95.36 - - [24/Feb/2008:00:12:59 +0100] "GET /burza/css/default.css?ciid=13 HTTP/1.1" 200 2516

193.198.16.211 - - [24/Feb/2008:00:16:43 +0100] "GET / HTTP/1.1" 200 15118

193.198.16.211 - - [24/Feb/2008:00:16:43 +0100] "GET /images/java-color100.png HTTP/1.1" 304 -

193.198.16.211 - - [24/Feb/2008:00:16:43 +0100] "GET /images/tomcat-1.png HTTP/1.1" 304 -

193.198.16.211 - - [24/Feb/2008:00:16:51 +0100] "GET /about.jsp?lang=hr HTTP/1.1" 200 4802

193.198.16.211 - - [24/Feb/2008:00:17:37 +0100] "GET /coursesTutorials.jsp?lang=hr HTTP/1.1" 200 6656

ltd...

RJEŠENJE

```
while(defined($redak = <>)){
@polja = split ' ', $redak;
$govno{$polja[0]}+=1;
}
```

```
foreach $racku (sort jabuka keys %govno){
print "$racku : $govno{$racku}\n";
}
```

```
sub jabuka {
return $govno{$b} <=> $govno{$a};
}
```

U log datoteci web poslužitelja zapisani su podaci o zahtjevima klijentskih računala. Podaci su navedeni sljedećim redom: IP adresa klijenta, dva polja s identifikatorima korisnika (u primjeru su ta polja prazna, odnosno označena znakom minus), polje s datumom i vremenom obrade zahtjeva (u uglatim zagradama), podaci o zahtjevu korisnika (unutar navodnika), oznaka statusa, i konačno veličina objekta vraćenog klijentu, izražena u bajtovima. Minus u pojedinom polju označava da podatak nije raspoloživ. Primjer log datoteke `localhost_access_log.txt`.

Napisati perl skriptu koja će učitati log datoteku koja se zadaje kao argument naredbenog retka, te za svakog klijenta odrediti broj prenesenih bajtova. Ispis sortirati silazno prema količini prenesenih podataka i oblikovati prema primjeru u nastavku.

Poziv skripte `zadatak4.pl`:

```
$ perl zadatak4.pl localhost_access_log.txt
```

```
78.3.194.174 : 166088
```

```
78.3.217.20 : 152404
```

```
89.164.244.106 : 137797
```

```
89.172.2.247 : 137797
```

```
66.249.73.45 : 99866
```

```
89.164.214.206 : 93186
```

```
78.1.101.126 : 91812
```

```
66.249.73.77 : 82225
```

```
83.131.36.70 : 47578
```

```
193.198.176.57 : 41926
```

```
193.198.16.211 : 36788
```

```
83.131.0.148 : 34819
```

```
89.172.53.0 : 34819
```

```
89.201.146.93 : 34819
```

```
89.164.203.68 : 33942
```

```
89.164.197.58 : 33445
```

```
193.198.83.95 : 30236
```

```
193.198.83.192 : 30236
```

```
89.164.165.147 : 28611
```

```
89.172.2.56 : 22295
```

```
193.198.17.121 : 19622
```

```
89.172.25.124 : 19438
```

```
89.172.7.242 : 19090
```

```
89.201.176.6 : 18965
```

```
78.0.95.36 : 18697
```

```
89.201.135.32 : 17809
```

```
78.134.144.219 : 17567
```

```
195.189.142.144 : 16342
```

```
82.193.208.247 : 16224
```

RJESENJE

```
while(<>){
    @polja = split /\s+/, $_;
    $ip_adresa = $polja[0];
    if ($polja[9] ne "-"){
        $brojbitova = $polja[9];
        $broj{$ip_adresa} += $brojbitova;
    }
}
foreach $ip (reverse sort {$broj{$a} <=> $broj{$b}} keys %broj){
    print "$ip : $broj{$ip}\n"
}
```

Uobičajen zapis podataka o znanstvenoj literaturi je bibtex. U prilogu je jedna takva datoteka (knjige.bib), s pojednostavljenim formatom. Koristi se samo jedna vrsta zapisa (@book), te samo 4 vrste polja (author, publisher, title i year). Potrebno je napisati Perl skriptu koja će na temelju bib datoteke čije se ime navodi kao argument naredbenog retka generirati (i ispisati na standardni izlaz) HTML kôd oblika (oznaka je šifra navedena na početku zapisa):

```
<UL>
<LI> autori, <I>title</I>, izdavač, godina (oznaka) </LI>
<LI> sljedeći rad </LI>
...
</UL>
```

Skripta treba provjeriti broj argumenata, te uspješnost otvaranja navedene datoteke s podacima. U slučaju neispravnog pozivanja ispisati uputu o načinu korištenja i izaći iz skripte.

Ispis oblikovati prema primjeru u nastavku.

Poziv skripte zadatak6.pl:

```
$ perl zadatak6.pl TestPrimjeri/knjige.bib
```

```
<UL>
<LI> R. O. Duda and P. E. Hart and D. G. Stork, <I>Pattern Classification</I>, Wiley, 2000 (DudaHart)</LI>
<LI> Linda G. Shapiro and George C. Stockman, <I>Computer Vision</I>, Prentice Hall, 2001 (shapiro01)</LI>
<LI> David A. Forsyth and Jean Ponce, <I>Computer Vision: A Modern Approach</I>, Prentice Hall, 2002
(Forsyth02)</LI>
<LI> R. Hartley and A. Zisserman, <I>Multiple View Geometry in Computer Vision</I>, Cambridge University
Press, 2004 (Hartley04)</LI>
<LI> J. R. Parker, <I>Algorithms for Image Processing and Computer Vision</I>, Wiley, 1996 (Parker96)</LI>
<LI> E. Trucco and A. Verri, <I>Introductory Techniques for 3-D Computer Vision</I>, Prentice Hall, 1998
(Trucco98)</LI>
</UL>
```

DATOTEKA

```
@book{DudaHart,
author= "R. O. Duda and P. E. Hart and D. G. Stork",
publisher= "Wiley",
title= "Pattern Classification",
year= "2000"
}
```

```
@book{shapiro01,
author = "Linda G. Shapiro and George C. Stockman",
publisher = "Prentice Hall",
title = "Computer Vision",
year = "2001"
}
```

```
@book{Forsyth02,
author = "David A. Forsyth and Jean Ponce",
title = "Computer Vision: A Modern Approach",
year = "2002",
```

```
publisher = "Prentice Hall"  
}
```

```
@book{Hartley04,  
author = "R. Hartley and A. Zisserman",  
title = "Multiple View Geometry in Computer Vision",  
year = "2004",  
publisher = "Cambridge University Press"  
}
```

```
@book{Parker96,  
author = "J. R. Parker",  
title = "Algorithms for Image Processing and Computer Vision",  
year = "1996",  
publisher = "Wiley"  
}
```

```
@book{Trucco98,  
AUTHOR = "E. Trucco and A. Verri",  
TITLE = "Introductory Techniques for 3-D Computer Vision",  
PUBLISHER = "Prentice Hall",  
YEAR = "1998"  
}
```

RJESENJE

```
f (@ARGV != 1){  
  die "Neispravan broj argumenata";  
}  
open READ, "$ARGV[0]";  
print "<UL>\n";
```

```
while (defined ($redak=<READ>)){  
  chomp($redak);  
  $redak =~ /\(@book.\)(.*)\./;  
  $book=$2;
```

```
  $redak=<READ>;  
  $redak =~ /\(")(.*)"/;  
  $author=$2;
```

```
  $redak=<READ>;  
  $redak =~ /\(")(.*)"/;  
  $publisher=$2;
```

```
  $redak=<READ>;  
  $redak =~ /\(")(.*)"/;  
  $title=$2;
```

```
  $redak=<READ>;  
  $redak =~ /\(")(.*)"/;  
  $year=$2;
```

```
# izbacuje znak { i prazan red
$redak=<READ>;
$redak=<READ>;

print "<LI> $author, </I>$title</I>, $publisher, $year ($book) </LI>\n";
}

print "<VUL>";
```


Napisati Perl skriptu koja priprema zapise za vremensku tablicu zadataka crontab. Ulazna datoteka se zadaje kao argument naredbenog retka, a primjer je dan u prilogu: grupe.txt. U datoteci je navedena informacija o laboratorijskim terminima: datum, vrijeme početka, vrijeme završetka i oznaka dvorane. Skripta treba za svaki termin (vremenski, neovisno o oznaci laboratorija) generirati 2 zapisa za crontab. Prvi zapis odgovara omogućavanju čitanja datoteke pola sata prije kraja laboratorijskog termina, a drugi onemogućavanju čitanja na kraju laboratorijskog termina. Format zapisa je (X označava redni broj termina):

```
minute sat dan mjesec * /bin/chmod 755 /www/htdocs/skrjez/zX
minute sat dan mjesec * /bin/chmod 700 /www/htdocs/skrjez/zX
```

Uočiti da se u crontab zapisu ne navodi godina, te da je zapis datuma u datoteci s grupama u obliku god-mjesec-dan. Generirani zapisi ispisuju se na standardni izlaz. Slobodno pretpostaviti da je kraj termina uvijek na puni sat.

Skripta treba provjeriti broj argumenata, te uspješnost otvaranja navedene datoteke s podacima. U slučaju neispravnog pozivanja ispisati uputu o načinu korištenja i izaći iz skripte.

Ispis oblikovati prema primjeru u nastavku.

Poziv skripte zadatak5.pl: \$ perl zadatak5.pl TestPrimjeri/grupe.txt

```
30 10 06 05 * /bin/chmod 755 /www/htdocs/skrjez/z1
00 11 06 05 * /bin/chmod 700 /www/htdocs/skrjez/z1
30 13 06 05 * /bin/chmod 755 /www/htdocs/skrjez/z2
00 14 06 05 * /bin/chmod 700 /www/htdocs/skrjez/z2
30 16 06 05 * /bin/chmod 755 /www/htdocs/skrjez/z3
00 17 06 05 * /bin/chmod 700 /www/htdocs/skrjez/z3
30 19 06 05 * /bin/chmod 755 /www/htdocs/skrjez/z4
00 20 06 05 * /bin/chmod 700 /www/htdocs/skrjez/z4
30 10 07 05 * /bin/chmod 755 /www/htdocs/skrjez/z5
00 11 07 05 * /bin/chmod 700 /www/htdocs/skrjez/z5
30 13 07 05 * /bin/chmod 755 /www/htdocs/skrjez/z6
00 14 07 05 * /bin/chmod 700 /www/htdocs/skrjez/z6
30 16 07 05 * /bin/chmod 755 /www/htdocs/skrjez/z7
00 17 07 05 * /bin/chmod 700 /www/htdocs/skrjez/z7
30 19 07 05 * /bin/chmod 755 /www/htdocs/skrjez/z8
00 20 07 05 * /bin/chmod 700 /www/htdocs/skrjez/z8
```

DATOTEKA

```
2010-05-06 08:00 11:00 PCLAB2
2010-05-06 08:00 11:00 PCLAB3
2010-05-06 11:00 14:00 PCLAB2
2010-05-06 14:00 17:00 PCLAB2
2010-05-06 17:00 20:00 PCLAB3
2010-05-07 08:00 11:00 A209
2010-05-07 08:00 11:00 A210
2010-05-07 08:00 11:00 PCLAB2
2010-05-07 11:00 14:00 A209
2010-05-07 11:00 14:00 A210
2010-05-07 11:00 14:00 PCLAB2 2
```

010-05-07 14:00 17:00 PCLAB2

2010-05-07 17:00 20:00 A209

2010-05-07 17:00 20:00 A210

RJESENJE

```
@a=<>;
$i=0;
$b="";
while(@a){
  chomp($a=shift(@a));
  @redak=split / /, $a;
  if ("${redak[0]}.${redak[1]}.${redak[2]}" eq "$b") { next;
}
$b="${redak[0]}.${redak[1]}.${redak[2]";
$i++;
$dat=$redak[0];
@dat=split /-/, $dat;
$dat1=@dat[1];
$dat2=@dat[2];
$vri2=@redak[2];
@vri2=split /:/, $vri2;
$vri2=@vri2[0];
$vri1=$vri2;
$vri2--;
print "30 $vri2 $dat2 $dat1 * /bin/chmod 755 /www/htdocs/skrjez/z$i\n";
print "00 $vri1 $dat2 $dat1 * /bin/chmod 700 /www/htdocs/skrjez/z$i\n";
}
```

Zadatak

U prilogu se nalazi datoteka [diplomski.txt](#) u kojoj su zapisani podaci o diplomskim radovima studenata. Uz ime i prezime studenta, naveden je naslov rada i godina u kojoj je rad obranjen. Napisati Perl skriptu koja će datoteku čije se ime zadaje kao argument naredbenog retka učitavati redak po redak, te na standardni izlaz zapisati HTML kôd u obliku:

```
<UL>
<LI> Ime Prezime, <I>Naslov</I>, godina </LI>
<LI> sljedeći rad </LI>
...
</UL>
```

Skripta treba provjeriti broj argumenata, te uspješnost otvaranja navedene datoteke s podacima. U slučaju neispravnog pozivanja ispisati uputu o načinu korištenja i izaći iz skripte.

Ispis oblikovati prema primjeru u nastavku.

Poziv skripte zadatak4.pl:

```
$ perl zadatak4.pl TestPrimjeri/diplomski.txt
```

```
<UL>
<LI> Maja Ključariček, <I>Procjena optičkog toka duž konture</I>, 2005 </LI>
<LI> Marina Hucaljuk, <I>Primjena Kalmanovog filtra u vizualnom praćenju objekata</I>, 2005 </LI>
<LI> Igor Mikša, <I>Učenje i raspoznavanje uzoraka u boji</I>, 2006 </LI>
<LI> Mladen Lombarović, <I>Detekcija osoba u infracrvenim slikama</I>, 2006 </LI>
<LI> Danijela Petrak, <I>Detekcija lica primjenom neuronskih mreža</I>, 2006 </LI>
...
</UL>
```

ovo je diplomski.txt:

```
Ključariček, Maja;Procjena optičkog toka duž konture;2005
Hucaljuk, Marina;Primjena Kalmanovog filtra u vizualnom praćenju objekata;2005
Mikša, Igor;Učenje i raspoznavanje uzoraka u boji;2006
Lombarović, Mladen;Detekcija osoba u infracrvenim slikama;2006
Petrak, Danijela;Detekcija lica primjenom neuronskih mreža;2006
Mijatović, Daniel;Sustav za segmentaciju slike pomakom prema srednjoj vrijednosti;2006
Košul, Luka;Praćenje objekata čestim filtrima;2006
Kuljić, Marko;Detekcija objekata deformabilnim predlojcima;2006
Lokin, Boško;Mjerenosti slikovnih uzoraka u boji;2006
Petrač, Deni;Lokalizacija objekata primjenom aktivnih modela oblika;2007
Ružan, Damir;Poravnavanje slika korištenjem modela usmjeravanja pisanje;2007
Stojević, Mislav;Uporaba značajki boje kao za detekciju lica;2007
Pokupec, Tomislav;Programski alat za praćenje rada i dijagnostiku složenih sustava;2007
Bagarić, Magdalena;Detekcija lica pomoću Gaborovih značajki;2008
Bobesić, Antonija;Primjena pravokutnih značajki za detekciju lica;2008
Boroš, Marko;Praćenje objekata u prometnim scenama;2008
Ošepo, Domagoj;Primjena strojnog učenja u detekciji osoba u slikama;2008
Čerovečki, Siniša;Automatizirano traženje sigurnosnih propusta u Web aplikacijama;2008
Oleković, Vidor;Sustav za pohranu i pretraživanje slika po tanskim maraka;2008
Dugandžić, Martina;Detekcija prometnih znakova u slikama u boji;2008
Grabar, Tatjana;Pretraživanje slikovnih baza temeljeno na lokalnim značajkama boje;2008
Svalina, Marin;Pretraživanje slikovnih baza temeljeno na značajkama tekture;2009
```

Valent, Tomislav;Analiza oblika kontura koA!tica maslina;2009

Zuban, Tadija;PraĀženje objekata adaptivnim pomicanjem prozora pretraA3ivanja;2009

RJESENJE

```
if(@ARGV<1){
```

```
    die "nedovoljni broj parametara";
```

```
}
```

```
if(!open DAT, "<$ARGV[0]") {
```

```
    print "datoteka se ne može otvoriti";
```

```
}
```

```
print "<UL>\n";
```

```
while($dat=<DAT>){
```

```
    @red=split(";", $dat);
```

```
    @ime=split(" ", $red[0]);
```

```
    print "<LI> ".$ime[0], $ime[1], "<I>";
```

```
    print $red[1].<I>, ".$red[2].<LI>".\n";
```

```
}
```

```
print '<\UL>'.\n";
```

Napomene

Skriptu imenovati zadatak1.pl

Argumenti: URL stranice, ime log datoteke

Izlaz: generirani podaci ispisuju se na stdout

Isprobanu skriptu predati kroz Ferko.

Obavezno zaključati upload!

Prilikom rješavanja izlaznog testa nije dozvoljeno koristiti Internet ni komunikaciju s kolegama!

Pri pisanju kôda voditi računa o preglednosti.

Zadatak

Napisati Perl skriptu koja će u log-datoteci web poslužitelja analizirati broj pristupa zadanoj stranici. Kao argumenti se navode ime stranice koja se analizira i ime log datoteke. Skripta treba ispisati broj pristupa s pojedinih IP adresa i to samo za one adrese s kojih je bilo barem 2 pristupa zadanoj stranici.

Skripta treba provjeriti broj argumenata, te uspješnost otvaranja navedene log datoteke. U slučaju neispravnog pozivanja ispisati uputu o načinu korištenja i izaći iz skripte.

Ispis oblikovati prema primjeru u nastavku, a za testiranje se može koristiti log-datoteka iz laboratorijske vježbe.

Poziv skripte zadatak1.pl:

```
$ perl zadatak1.pl "/zadaci-faq.html" TestPrimjeri/localhost_access_log.2008-02-24.txt
```

```
-----  
Broj pristupa stranici: /zadaci-faq.html
```

```
IP adresa pristupatelja : Broj pristupa
```

```
-----  
89.164.7.174 : 8  
89.172.20.163 : 2  
78.0.25.238 : 2  
89.164.46.53 : 9  
83.131.54.227 : 2  
89.172.56.25 : 2  
90.157.251.180 : 2  
83.131.40.124 : 2  
89.164.250.5 : 15  
90.157.248.22 : 2  
89.172.2.247 : 2  
78.1.111.73 : 2  
78.1.106.36 : 2  
89.172.53.0 : 2  
193.198.16.211 : 13  
89.164.228.250 : 3  
-----
```

RJESENJE

```
#!/perl
```

```
$niz = shift @ARGV;
```

```
@sve = <>;
```

```
chomp($niz);
```

```

foreach $redak (@sve)
{
    $_ = $redak;
    if(m/^. *GET $niz .*/)
    {
        @pomocna = split /\ /, $redak;
        $ip = $pomocna[0];
        $rezultat{$ip} +=1;
        #print $ip . " : " . $rezultat{$ip} . "\n";
    }
}

@kljucevi = %rezultat;
$i = 0;
$zadnji = $#kljucevi + 1;
while($i < $zadnji){
    if ($i % 2 != 0){
        $i += 1;
    }else{
        if($kljucevi[$i+1] >= 2){
            print $kljucevi[$i] . " : " . $kljucevi[$i+1] . "\n";
        }
        $i += 1;
    }
}

```

ZADATAK

U log datoteci web poslužitelja zapisani su podaci o zahtjevima klijentskih računala. Podaci su navedeni sljedećim redom: IP adresa klijenta, dva polja s identifikatorima korisnika (u primjeru su ta polja prazna, odnosno označena znakom minus), polje s datumom i vremenom obrade zahtjeva (u uglatim zagradama), podaci o zahtjevu korisnika (unutar navodnika), oznaka statusa, i konačno veličina objekta vraćenog klijentu, izražena u bajtovima. Minus u pojedinom polju označava da podatak nije raspoloživ. Primjer log datoteke je:

```
89.172.5.200 - - [24/Feb/2008:00:00:21 +0100] "GET /burza/b/Logout.action?ciid=26 HTTP/1.1" 302 -
89.172.5.200 - - [24/Feb/2008:00:00:21 +0100] "GET /burza/b/Main.action HTTP/1.1" 200 1111
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET / HTTP/1.1" 200 15118
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /images/java-color100.png HTTP/1.1" 200 11698
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /images/tomcat-1.png HTTP/1.1" 200 5641
78.3.194.174 - - [24/Feb/2008:00:03:44 +0100] "GET /favicon.ico HTTP/1.1" 404 988
78.3.194.174 - - [24/Feb/2008:00:04:34 +0100] "HEAD /zadaci.pdf HTTP/1.1" 200 -
78.3.194.174 - - [24/Feb/2008:00:04:35 +0100] "GET /zadaci.pdf HTTP/1.1" 200 86872
78.3.194.174 - - [24/Feb/2008:00:04:39 +0100] "GET /about.jsp?lang=hr HTTP/1.1" 200 4802
78.3.194.174 - - [24/Feb/2008:00:04:42 +0100] "GET /members.jsp?lang=hr HTTP/1.1" 200 7260
78.3.194.174 - - [24/Feb/2008:00:04:48 +0100] "GET /challenges.jsp?lang=hr HTTP/1.1" 200 10561
```

Napisati perl skriptu koja će učitati log datoteku koja se zadaje kao argument naredbenog retka, te za svakog klijenta izbrojati pristupe poslužitelju. Ispis sortirati silazno prema broju pristupa i oblikovati prema primjeru u nastavku.

Poziv skripte zadatak9.pl:

```
$ perl zadatak9.pl localhost_access_log.txt 83.131.36.70 : 21 193.198.176.57 : 19 89.164.203.68 : 17
193.198.17.121 : 14 78.3.194.174 : 13 89.172.25.124 : 13 89.172.7.242 : 12 89.164.165.147 : 12 89.172.2.56 : 12
193.198.140.56 : 12 78.134.144.219 : 10 193.198.16.211 : 10 78.0.95.36 : 9 195.189.142.144 : 9 82.193.208.247 : 9
78.0.125.72 : 9 89.201.135.32 : 9 66.249.73.45 : 8 89.164.244.106 : 8 78.3.217.20 : 8 89.172.2.247 : 8
89.164.214.206 : 7 89.172.53.0 : 6 78.1.101.126 : 6 66.249.73.77 : 5 193.198.17.88 : 5 83.131.0.148 : 5
89.201.146.93 : 5 89.164.197.58 : 4 193.198.83.192 : 4 89.172.5.241 : 3 193.198.83.95 : 3 89.201.176.6 : 3
89.172.20.197 : 2 89.172.37.151 : 2 89.172.5.200 : 2 78.1.99.219 : 2
```

novi redak fali, c/p...

RJESENJE:

```
#!/usr/bin/perl
```

```
if ( $#ARGV == -1 ) {

    die ("ERROR: Please set file to open.\n");

}

foreach $File ( @ARGV ) {

    if ( ! open LOG_FILE, $File ) {

        die ( "ERROR: Can't open file.\nDESCRIPTION: $!\n" );

    }

    while ( $Line = <LOG_FILE> ) {

        @Parts = split ( " ", $Line);

        $Hours{$Parts[0]} += 1;

    }

    foreach $Key (sort { $Hours{$a} cmp $Hours{$b} } keys %Hours) {

        printf ( "%-16s : %d\n", $Key, $Hours{$Key});

        delete($Hours{$Key});          # RESET

    }

}
```