

CSCI 132 Practical Unix Programming

Homework 5

Solution

Saad Mneimneh
Computer Science
Hunter College of CUNY

PART 1

Consider a file that consisting of lines that contain a color and an amount separated by colons, like:

```
red:27
yellow:102
green:311
yellow:12
blue:45
```

Write a program that calculates the total amount for each color. The final program, for the above example file, should output (not necessarily in that order):

```
red:27
yellow:114
green:311
blue:45
```

Hint: use a hash data structure where the keys are the colors, and the values are the amounts.

Hint: use the match operator on each line to isolate the color and the amount. You can also use the split function with `:` being the delimiter.

ANSWER:

```
#!/usr/bin/perl

use strict;

#declare a hash variable
my %hash;

#read the file given at the prompt
while (<>) {
    #for each line, chomp and split it
    chomp($_);
    my @a=split(/:/,$_);
```

```

    #if hash key already exists, update value
    if ($hash{$a[0]}) {
        $hash{$a[0]}=$hash{$a[0]}+$a[1];
    }
    else {
        $hash{$a[0]}=$a[1];
    }
}

#print keys and values
foreach my $k (keys(%hash)) {
    my $v=$hash{$k};
    print "$k:$v\n";
}

```

Another solution using the match operator instead of split:

```

#!/usr/bin/perl

use strict;

#declare a hash variable
my %hash;

#read the file given at the prompt
while (<>) {
    #for each line, chomp and match
    chomp($_);
    $_ =~ m/(.*):(.*)/;
    #if hash key already exists, update value
    if ($hash{$1}) {
        $hash{$1}=$hash{$1}+$2;
    }
    else {
        $hash{$1}=$2;
    }
}

#print keys and values
foreach my $k (keys(%hash)) {
    my $v=$hash{$k};
    print "$k:$v\n";
}

```

PART 2

Consider a file that consists of lines like the following:

```
r 1 y 2 g 3 g 4 y 2 r 4 r 3 y 5 y 3 g 3 g 6 g 2 r 1
```

We would like to merge all amounts with the same tag together, **while preserving the order**. For instance, the following line should become:

```
r 1 y 2 g 7 y 2 r 7 y 8 g 11 r 1
```

Write a program to do that.

Hint: use split and arrays, and check for consecutive tags.

ANSWER: The line is assumed to be the string s. Code for reading lines from the file is not shown.

```
chomp($s);
my @a=split(/ /, $s);

my $prev; #keep track of the previous color seen
my @b=(); #this will be the array with the new values (shorter than a)
for(my $i=0; $i< $#a; $i=$i+2) {
    if ($a[$i] eq $prev) {
        $b[$#b]=$b[$#b]+$a[$i+1]; #update last value in b
    }
    else {
        push(@b, $a[$i]); #push new pair of color/value
        push(@b, $a[$i+1]);
        $prev=$a[$i];
    }
}
}
$s = "@b\n";
print $s;
```

The following code shows a solution using the match operator.

```
while ($s =~ m/([ryg]) ([0-9]*) \1 ([0-9]*)/) {
    my $r=$2+$3;
    $s =~ s/([ryg]) [0-9]* \1 [0-9]*/$1 $r/;
}
print $s;
```

PART 3

Write your own cat command. You should be able to use it as follows (examples)

```
cat -h3 filename    displays the first 3 lines
cat -t5 filename    displays the last 5 lines
cat filename         displays the entire file
```

Hint: what you type at the command line will be in @ARGV. So try to match the first element with something that starts with dash followed by a letter, followed by a number. If so, do what you have to do.

Hint: After you determine what the first element of @ARGV is, you have to remove it from @ARGV if it is not the file name, so that <> will work property.

ANSWER:

```
#!/usr/bin/perl

use strict;

my $headortail='x';
my $number;

if ($ARGV[0] =~ m/-([ht])([0-9]*)/) {
    $headortail=$1;
    $number=$2;
    shift(@ARGV); #remove first element
}

while (open (FILE, $ARGV[0])) {
    my @a=<FILE>; #assume entire file can be loaded in memory
    if ($headortail eq 'h') {
        my $end;
        if ($number-1>$#a) { #if beyond the last line
            $end=$#a;
        }
        else {
            $end=$number-1;
        }
        for (my $i=0; $i<=$end; $i=$i+1) {
            print $a[$i];
        }
    }
    elsif ($headortail eq 't') {
        my $start=$#a-$number+1;
        if ($start<0) {
            $start=0;
        }
        for (my $i=$start; $i<=$#a; $i=$i+1) {
            print $a[$i];
        }
    }
    else {
        for (my $i=0; $i<=$#a; $i=$i+1) {
            print $a[$i];
        }
    }
    shift(@ARGV);
}
```