COMMANDS

Tuesday, 8 June 2021 2:38 PM

1) Multiple patterns and actions:

```
awk '/up/{print "UP: " NF, $0} /down/{print "DOWN: ", NF, $0} ' dukeofyork.txt
```

2) Specify regular expression as a field separator:

```
awk -F '[,!]' '{print NR,$2}'
```

3) Record and field separators:

```
awk 'BEGIN{FS=","; RS="!"}{print $3}' onebigline.txt
```

```
three
six
nine
twelve
```

1)
```
awk 'BEGIN{RS=""; FS="\n"} {name=$1; address=$2; statezip=$3}
multiaddress.txt
```

```
Gladys Rim, 322 New Horizon Blvd, Milwaukee, WI 53207
Yuki Whobrey, 1 State Route 27, Taylor, MI 48180
Fletcher Flosi, 394 Manchester Blvd, Rockford, IL 61109
Bette Nicka, 6 S 33rd St, Aston, PA 19014
Veronika Inouye, 6 Greenleaf Ave, San Jose, CA 95111
Willard Kolmetz, 618 W Yakima Ave, Irving, TX 75062
```

```
{print name", " address ", " statezip}'
```

OUTPUT FIELD SEPARATORS AND OUTPUT RECORD SEPARAT

```
 1)  awk 'BEGIN{FS=","; OFS="\t"} {print $1, $2, $3} ' nameemailav
Art Venere art@venere.org 256.62394383
Lenna Paprocki lpaprocki@hotmail.com 259.688783099
Donette Foller donette.foller@cox.net 282.32979844
Simona Morasca simona@morasca.com 139.51911647
Dominque Dickerson dominque.dickerson@dickerson.org 167.045197551
Leota Dilliard leota@hotmail.com 253.646335223
Sage Wieser sage_wieser@cox.net 155.55576823
Kris Cho kris@gmail.com 210.794277775
Minna Amigon minna_amigon@yahoo.com 95.47955397
Abel Maclead amaclead@gmail.com 225.774477397
Kiley Caldarera kiley.caldarera@aol.com 172.957628871
Graciela Ruta gruta@cox.net 202.68364784
Josephine Darakjy josephine_darakjy@darakjy.org 178.877840188
Cammy Albares calbares@gmail.com 290.446513401
Mattie Poquette mattie@aol.com 283.23995223
Meaghan Garufi meaghan@hotmail.com 227.142916195
Gladys Rim gladys.rim@rim.org 243.459635712
Yuki Whobrey yuki_whobrey@aol.com 128.321717297
Fletcher Flosi fletcher.flosi@yahoo.com 221.394141603
```

OR

```
awk -F, '{print $1 "\t" $2 "\t" $3}' nameemailavg.csv
```

REGULAR EXPRESSIONS:

```
 A)  awk '$4 ~ /up/{print $0}' dukeofyork.txt  : 4th field cor

He marched them up to the top of the hill
They were neither up nor down
```

TORS

```
vg.csv
```

```
ntains regular expression pattern /up/
```

```
B) awk '$3 ~ /the/{print}' dukeofyork.txt  : 3rd field matche
He marched them up to the top of the hill
And when they were up they were up
And when they were down they were down
And when they were only half-way up
They were neither up nor down
```

```
 C)  awk '$3 ~ /^the/{print}' dukeofyork.txt  : 3rd field begi
     He marched them up to the top of the hill
     And when they were up they were up
     And when they were down they were down
     And when they were only half-way up
```

```
D) awk '/a[a-zA-Z]*c/{print $0}' : Match 0 or more occurrence

abbddbBc
abbddbBc
abbbbbdssbbC
abHFYEHSHc
abHFYEHSHc
```

```
 E)  awk '/a[a-zA-Z]{3}c/{print $0}': Exactly matches 3 occurr
     abc
     abbc
     aABCc
     aABCc
     aZDbc
     aZDbc
```

```
 F)  Grouping characters: awk '/(anish){2}[a-z]c/{print $0}'.
anishanishzc
anishanishzc
anishbc
```

es /the/

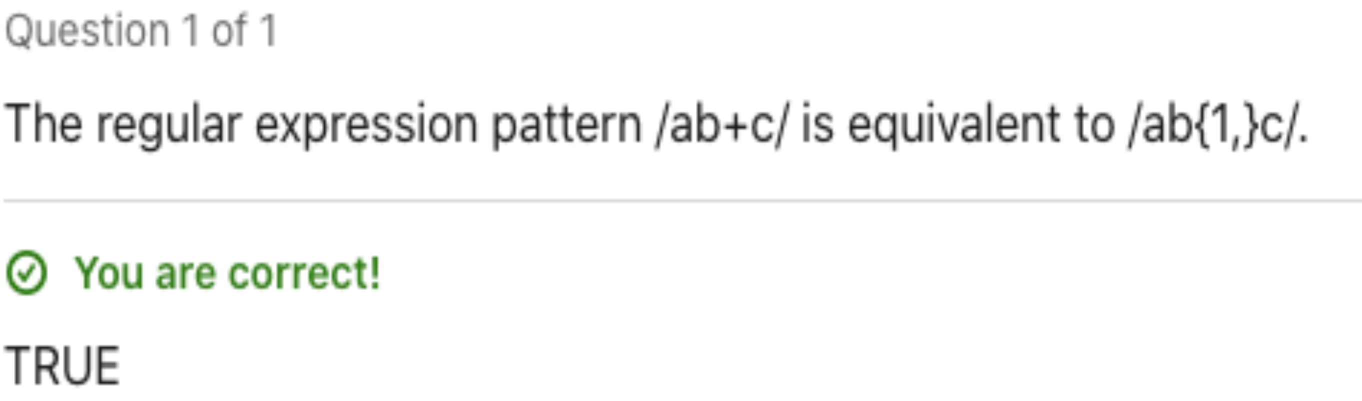
ins with /the/

es of the char set pattern in between i.e [a-zA-Z]*:

rences of the pattern [a-zA-Z] in between:

: Exactly.2 occurrences of pattern (anish):

```
anishbc
```



Question 1 of 1

The regular expression pattern /ab+c/ is equivalent to /ab{1,}c/.

✓ You are correct!

TRUE

CONTROL STRUCTURES

```
A)  awk '{ if ( NF < 8) {print "Short Line:", $0 } else { pri
    Files/dukeofyork.txt


    Short Line: The grand old Duke of York
    Short Line: He had ten thousand men
    Long Line:  He marched them up to the top of the hill
    Short Line: And he marched them down again
    Long Line:  And when they were up they were up
    Long Line:  And when they were down they were down
    Short Line: And when they were only half-way up
    Short Line: They were neither up nor down
```

OR We can create an AWK file and add the code inside it us

```
int "Long Line: ", $0 } }' Exercise\
```

```
ing -f flag:
    awk -f condition.awk dukeofyork.txt

    SHORT LINE: The grand old Duke of York
    SHORT LINE: He had ten thousand men
    LONG LINE: He marched them up to the top of the hill
    SHORT LINE: And he marched them down again
    SHORT LINE: And when they were up they were up
    SHORT LINE: And when they were down they were down
    SHORT LINE: And when they were only half-way up
    SHORT LINE: They were neither up nor down
```

The AWK File condition.awk:

```
{
        if (NF > 8)
        {
                print("LONG LINE:", $0)
        }
        else
        {
                print("SHORT LINE:", $0)
        }

}
```

B) FOR LOOP:

```
    {
            for  ( i = 1 ; i<=3; i++)
            {
                    print "Line ", NR ",field ", i ": ", $i ;
            }

    }

    awk -f forloop.awk dukeofyork.txt

    Line  1,field  1:  The
    Line  1,field  2:  grand
```

```
Line  1,field  2:  grand
Line  1,field  3:  old
Line  2,field  1:  He
Line  2,field  2:  had
Line  2,field  3:  ten
Line  3,field  1:  He
Line  3,field  2:  marched
Line  3,field  3:  them
Line  4,field  1:  And
Line  4,field  2:  he
Line  4,field  3:  marched
Line  5,field  1:  And
Line  5,field  2:  when
Line  5,field  3:  they
Line  6,field  1:  And
Line  6,field  2:  when
Line  6,field  3:  they
Line  7,field  1:  And
Line  7,field  2:  when
Line  7,field  3:  they
Line  8,field  1:  They
Line  8,field  2:  were
Line  8,field  3:  neither
```

FORMATTING THER OUTPUT– USING P

A) `awk -F, '{printf("%s\t%s\t%d\n", $1, $2, $3)}' nameemaila`

```
Art Venere	art@venere.org	256
Lenna Paprocki	lpaprocki@hotmail.com	259
Donette Foller	donette.foller@cox.net	282
Simona Morasca	simona@morasca.com	139
Dominque Dickerson	dominque.dickerson@dickerson.org
Leota Dilliard	leota@hotmail.com	253
```

RINTF

avg.csv

167

```
Sage Wieser     sage_wieser@cox.net   155
Kris Cho   kris@gmail.com  210
Minna Amigon    minna_amigon@yahoo.com    95
Abel Maclead    amaclead@gmail.com   225
Kiley Caldarera kiley.caldarera@aol.com    172
Graciela Ruta   gruta@cox.net   202
Josephine Darakjy    josephine_darakjy@darakjy.org   178
Cammy Albares   calbares@gmail.com   290
Mattie Poquette mattie@aol.com  283
Meaghan Garufi  meaghan@hotmail.com  227
Gladys Rim gladys.rim@rim.org   243
Yuki Whobrey    yuki_whobrey@aol.com 128
Fletcher Flosi  fletcher.flosi@yahoo.com   221
```

B) Formatting output with width and precision specifiers:

```
awk -F, '{printf("%-20s %-35s %3f\n",$1,$2,$3)}' nameemai

Art Venere           art@venere.org
Lenna Paprocki       lpaprocki@hotmail.com
Donette Foller       donette.foller@cox.net
Simona Morasca       simona@morasca.com
Dominque Dickerson   dominque.dickerson@dickerson.org
Leota Dilliard       leota@hotmail.com
Sage Wieser          sage_wieser@cox.net
Kris Cho             kris@gmail.com
Minna Amigon         minna_amigon@yahoo.com
Abel Maclead         amaclead@gmail.com
Kiley Caldarera      kiley.caldarera@aol.com
Graciela Ruta        gruta@cox.net
Josephine Darakjy    josephine_darakjy@darakjy.org
Cammy Albares        calbares@gmail.com
Mattie Poquette      mattie@aol.com
Meaghan Garufi       meaghan@hotmail.com
Gladys Rim           gladys.rim@rim.org
Yuki Whobrey         yuki_whobrey@aol.com
Fletcher Flosi       fletcher.flosi@yahoo.com
```

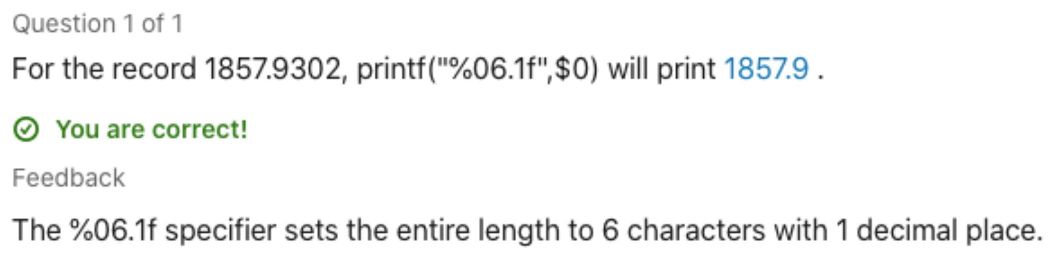
```
ilavg.csv

 256.623944
 259.688783
 282.329798
 139.519116
 167.045198
 253.646335
 155.555768
 210.794278
 95.479554
 225.774477
 172.957629
 202.683648
 178.877840
 290.446513
 283.239952
 227.142916
 243.459636
 128.321717
 221.394142
```

C) Width and precision specifier for decimal values: % <width

```
awk -F, '{printf("%-20s %-35s %6.2f\n",$1,$2,$3)}' nameem

Art Venere             art@venere.org
Lenna Paprocki         lpaprocki@hotmail.com
Donette Foller         donette.foller@cox.net
Simona Morasca         simona@morasca.com
Dominque Dickerson     dominque.dickerson@dickerson.org
Leota Dilliard         leota@hotmail.com
Sage Wieser            sage_wieser@cox.net
Kris Cho               kris@gmail.com
Minna Amigon           minna_amigon@yahoo.com
Abel Maclead           amaclead@gmail.com
Kiley Caldarera        kiley.caldarera@aol.com
Graciela Ruta          gruta@cox.net
Josephine Darakjy      josephine_darakjy@darakjy.org
Cammy Albares          calbares@gmail.com
Mattie Poquette        mattie@aol.com
Meaghan Garufi         meaghan@hotmail.com
Gladys Rim             gladys.rim@rim.org
Yuki Whobrey           yuki_whobrey@aol.com
Fletcher Flosi         fletcher.flosi@yahoo.com
```



Question 1 of 1

For the record 1857.9302, printf("%06.1f",$0) will print 1857.9 .

You are correct!

Feedback

The %06.1f specifier sets the entire length to 6 characters with 1 decimal place.

STRING MANIPULATION USING INB

```
/length> . <precision/no of digits after decimal>f:

ailavg.csv

 256.62
 259.69
 282.33
 139.52
 167.05
 253.65
 155.56
 210.79
  95.48
 225.77
 172.96
 202.68
 178.88
 290.45
 283.24
 227.14
 243.46
 128.32
 221.39
```

UILD FUNCTIONS

A) Using sub(/regexp/,<replace_pattern_with>) function to rep

```
    awk '{sub(/the/,""); print}' dukeofyork.txt

    The grand old Duke of York
    He had ten thousand men
    He marched m up to the top of the hill
    And he marched m down again
    And when y were up they were up
    And when y were down they were down
    And when y were only half-way up
    They were neir up nor down
```

B) Another example of using sub() function to replace a strin

```
    awk '{sub(/the[ym]/,"ALL OF THEM"); print}' dukeofyork.tx

    The grand old Duke of York
    He had ten thousand men
    He marched ALL OF THEM up to the top of the hill
    And he marched ALL OF THEM down again
    And when ALL OF THEM were up they were up
    And when ALL OF THEM were down they were down
    And when ALL OF THEM were only half-way up
    They were neither up nor down
```

C) Gsub: Global changing of all matching patterns everywhere

```
awk '{gsub(/the[ym]/,"ALL OF THEM"); print}' dukeofyork.txt

The grand old Duke of York
```

lace a first matching string pattern :

g pattern:

xt

in the input:

```
He had ten thousand men
He marched ALL OF THEM up to the top of the hill
And he marched ALL OF THEM down again
And when ALL OF THEM were up ALL OF THEM were up
And when ALL OF THEM were down ALL OF THEM were down
And when ALL OF THEM were only half-way up
They were neither up nor down
```

ARRAYS

A) Associative arrays:

```
awk '{a["first"]=$1; a["second"]=$2 ; a["third"]=$3; print a[

one two three
three two one
```

B) Using a for loop to traverse the arrray:

```
    awk '{a["first"]=$1; a["second"]=$2 ; a["third"]=$3; for(


    one two three four
    first one
    third three
    second two
```

C) Number of occurrences of each distinct word in an input:

```
    awk -f count_words.awk dukeofyork.txt  | sort -rn  -k 2
```

```
["third"], a["second"], a["first"]}'
```

```
(i in a) {print i, a[i] }}'
```

```
were 6
they 6
up 5
down 4
and 4
when 3
the 3
he 3
them 2
of 2
marched 2
york 1
top 1
to 1
thousand 1
ten 1
only 1
old 1
nor 1
neither 1
men 1
hill 1
half-way 1
had 1
grand 1
duke 1
again 1
```

THE AWK PROGRAM: count_words.awk

```
{
        for( i=1 ; i<=NF; i++) {
                words[tolower($i)]++;
        }


}

END {
        for (i in words) {
```

```
                print i, words[i];
        }

}
```

PIPES

A) Printing total size of .docx files in a directory:

```
ls -l | awk '/\.docx$/{total += $5} END{print total}'

2399490
```

B) All the AWK files and their sizes from a file ls.out

```
awk '/.awk*/{total += $5; print } END{print total}' ls.ou

-rwxr-xr-x@ 1 anish.walia  staff    281 Jun  7 16:33 awk.
-rw-r--r--  1 anish.walia  staff     89 Aug 18 13:53 conc
-rw-r--r--  1 anish.walia  staff    114 Aug 30 18:53 cour
-rwxr-xr-x@ 1 anish.walia  staff  10886 Jun  7 16:33 csv.
-rwxr-xr-x@ 1 anish.walia  staff    207 Jun  7 16:33 exce
-rwxr-xr-x@ 1 anish.walia  staff     79 Jun  7 16:33 firs
-rw-r--r--  1 anish.walia  staff     82 Aug 18 13:59 forl
-rwxr-xr-x@ 1 anish.walia  staff    370 Jun  7 16:33 join
-rwxr-xr-x@ 1 anish.walia  staff    154 Jun  7 16:33 remo
-rwxr-xr-x@ 1 anish.walia  staff    518 Jun  7 16:33 scor
-rwxr-xr-x@ 1 anish.walia  staff     88 Jun  7 16:33 shor
-rwxr-xr-x@ 1 anish.walia  staff    241 Jun  7 16:33 tran
-rwxr-xr-x@ 1 anish.walia  staff    112 Jun  7 16:33 word
```

```
ut

.out*
dition.awk
nt_words.awk
.awk*
el_final.awk*
stthree.awk*
loop.awk
n.awk*
ovethe.awk*
res_final.awk*
rtlong.awk*
nspose.awk*
dusage.awk*
```

```
13221
```

C) Using sed to first replace all .awk* files to .awk:

```
sed 's/\*$//' ls.out |   awk '/.awk*/{total += $5; print

-rwxr-xr-x@ 1 anish.walia  staff    281 Jun  7 16:33 awk.
-rw-r--r--  1 anish.walia  staff     89 Aug 18 13:53 conc
-rw-r--r--  1 anish.walia  staff    114 Aug 30 18:53 coun
-rwxr-xr-x@ 1 anish.walia  staff  10886 Jun  7 16:33 csv.
-rwxr-xr-x@ 1 anish.walia  staff    207 Jun  7 16:33 exce
-rwxr-xr-x@ 1 anish.walia  staff     79 Jun  7 16:33 firs
-rw-r--r--  1 anish.walia  staff     82 Aug 18 13:59 forl
-rwxr-xr-x@ 1 anish.walia  staff    370 Jun  7 16:33 join
-rwxr-xr-x@ 1 anish.walia  staff    154 Jun  7 16:33 remo
-rwxr-xr-x@ 1 anish.walia  staff    518 Jun  7 16:33 scor
-rwxr-xr-x@ 1 anish.walia  staff     88 Jun  7 16:33 shor
-rwxr-xr-x@ 1 anish.walia  staff    241 Jun  7 16:33 tran
-rwxr-xr-x@ 1 anish.walia  staff    112 Jun  7 16:33 word
13221
```

D) Pre-processing a AWK program with other:

```
awk '{print $1,  $2 , "<" $3 ">"}' nicknames.txt

Alex Loader <alex@loader.com>
Lashon Vizarro <lashon@aol.com>
Lauran Burnard <lburnard@burnard.com>
Ceola Setter <ceola.setter@setter.org>
My Rantanen <my@hotmail.com>
Lorrine Worlds <lorrine.worlds@worlds.com>
Peggie Sturiale <peggie@cox.net>
Marvel Raymo <mraymo@yahoo.com>
Daron Dinos <daron_dinos@cox.net>
An Fritz <an_fritz@hotmail.com>
Portia Stimmel <portia.stimmel@aol.com>
Rhea Aredondo <rhea.aredondo@cox.net>
```

```
} END{print total}'

.out
dition.awk
nt_words.awk
.awk
el_final.awk
stthree.awk
loop.awk
n.awk
ovethe.awk
res_final.awk
rtlong.awk
nspose.awk
dusage.awk
```

```
Rhea Aredondo <rhea_aredondo@cox.net>
Benedict Sama <bsama@cox.net>
Alyce Arias <alyce@arias.org>
Heike Berganza <heike@gmail.com>
Carey Dopico <carey_dopico@dopico.org>
Dottie Hellickson <dottie@hellickson.org>
Deandrea Hughey <deandrea@yahoo.com>
Kimberlie Duenas <kimberlie_duenas@yahoo.com>
Martina Staback <martina_staback@staback.com>
Skye Fillingim <skye_fillingim@yahoo.com>
Jade Farrar <jade.farrar@yahoo.com>
Charlene Hamilton <charlene.hamilton@hotmail.com>
Geoffrey Acey <geoffrey@gmail.com>
Stevie Westerbeck <stevie.westerbeck@yahoo.com>
Pamella Fortino <pamella@fortino.com>
Harrison Haufler <hhaufler@hotmail.com>
Johnna "Zip" <Engelberg>
Buddy Cloney <buddy.cloney@yahoo.com>
Dalene Riden <dalene.riden@aol.com>
Jerry "Killer" <Zurcher>
Haydee Denooyer <hdenooyer@denooyer.org>
Joseph Cryer <joseph_cryer@cox.net>
Deonna Kippley <deonna_kippley@hotmail.com>
Raymon Calvaresi <raymon.calvaresi@gmail.com>
Alecia Bubash <alecia@aol.com>
```

The above 2 columns have one extra field of nicknames in them,

E) Pre-processing 2 AWK programs: In the below we intentional

```
awk '{if(NF==4){$2=""}; print}' nicknames.txt | awk '{pri
```

```
Alex Loader <alex@loader.com>
Lashon Vizarro <lashon@aol.com>
Lauran Burnard <lburnard@burnard.com>
Ceola Setter <ceola.setter@setter.org>
My Rantanen <my@hotmail.com>
Lorrine Worlds <lorrine.worlds@worlds.com>
Peggie Sturiale <peggie@cox.net>
```

```
 which is the 2nd column


lly set the 2nd field to empty

int $1 , $2, "<" $3 ">"}'
```

```
Marvel Raymo <mraymo@yahoo.com>
Daron Dinos <daron_dinos@cox.net>
An Fritz <an_fritz@hotmail.com>
Portia Stimmel <portia.stimmel@aol.com>
Rhea Aredondo <rhea_aredondo@cox.net>
Benedict Sama <bsama@cox.net>
Alyce Arias <alyce@arias.org>
Heike Berganza <heike@gmail.com>
Carey Dopico <carey_dopico@dopico.org>
Dottie Hellickson <dottie@hellickson.org>
Deandrea Hughey <deandrea@yahoo.com>
Kimberlie Duenas <kimberlie_duenas@yahoo.com>
Martina Staback <martina_staback@staback.com>
Skye Fillingim <skye_fillingim@yahoo.com>
Jade Farrar <jade.farrar@yahoo.com>
Charlene Hamilton <charlene.hamilton@hotmail.com>
Geoffrey Acey <geoffrey@gmail.com>
Stevie Westerbeck <stevie.westerbeck@yahoo.com>
Pamella Fortino <pamella@fortino.com>
Harrison Haufler <hhaufler@hotmail.com>
Johnna Engelberg <jlengelberg@engelberg.org>
Buddy Cloney <buddy.cloney@yahoo.com>
Dalene Riden <dalene.riden@aol.com>
Jerry Zurcher <jzurcher@zurcher.org>
Haydee Denooyer <hdenooyer@denooyer.org>
Joseph Cryer <joseph_cryer@cox.net>
Deonna Kippley <deonna_kippley@hotmail.com>
Raymon Calvaresi <raymon.calvaresi@gmail.com>
Alecia Bubash <alecia@aol.com>
```