
Da colonna a riga

- ❖ Dato il file di testo **tabella.txt** così fatto:

Luigi

Marco

Giovanni

Luigi

Giorgio

Luca

Maria

- ❖ scrivere uno script che eseguito da **AWK** sul file stampa il contenuto del file tutto su di una riga:

Luigi Marco Giovanni Luigi Giorgio Luca Maria

Da colonna a riga (sol.)

❖ Soluzione:

awk

```
'BEGIN {RS="\n"; ORS=" ";print "\n"} {print $0} END{print "\n\n"}'
```

tabella.txt

Leggi fino a dove dico io

- ❖ Dato il file di testo **poesia.txt** così fatto:

La donzelletta vien dalla campagna,
In sul calar del sole,
Col suo fascio dell'erba; e reca in mano
Un mazzolin di rose e di viole,
Onde, siccome suole,
Ornare ella si appresta
Dimani, al dì di festa, il petto e il **crine**.
Siede con le vicine
Su la scala a filar la vecchierella
Incontro là dove si perde il giorno;
E novellando vien del suo buon tempo,
Quando ai dì della festa ella si omava,
Ed ancor sana e snella
Solea danzar la sera intra di quei
Ch'ebbe compagni dell'età più bella.

- ❖ scrivere uno script che eseguito da **AWK** sul file stampa il contenuto del file fino alla stringa "**crine**":

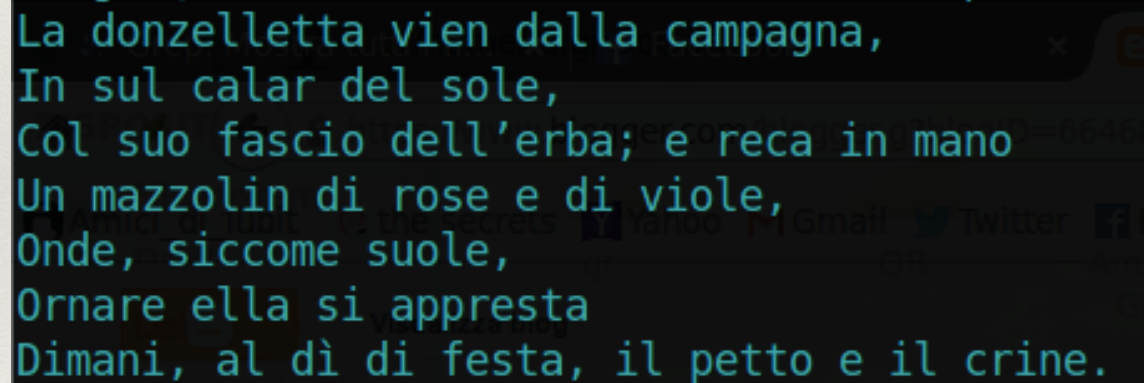
Leggi fino a dove dico io (sol.)

❖ Soluzione:

`awk`

`'{print} /crine/ {exit}'`

`poesia.txt`



```
La donzelletta vien dalla campagna,  
In sul calar del sole,  
Col suo fascio dell'erba; e reca in mano  
Un mazzolin di rose e di viole,  
Onde, siccome suole,  
Ornare ella si appresta  
Dimani, al dì di festa, il petto e il crine.
```

Quante volte ricorre una determinata parola in un testo

- ❖ Dato il file di testo **tabella.txt** così fatto:

Luigi;Marco;Nino;Nicola;Alberto;Luigi;

Marco;Giovanni;Giorgio;Maria;

Giovanni;Ottavio;Luigi;Luigi;Nino;

Luigi;Guglielmo;Nino;Ennio;Luigi;

Giorgio;Vittorio;

Luca;Marta;Maria;Luigi;Salvo;

Maria;Nino;Ennio;Luigi;Maria;

- ❖ scrivere un file programma che eseguito da **AWK** sul file determini quante volte è contenuto il nome "Luigi" nel file **"tabella.txt"**:

Nel file tabella.txt il nome Luigi ricorre 8 volte

Quante volte ricorre una determinata parola in un testo (sol.)

❖ Soluzione con comando Linux

```
grep -o Luigi tabella.txt | wc -l
```

Quante volte ricorre una determinata parola in un testo (sol.)

- ❖ Soluzione con file programma AWK (es.awk)

```
#!/usr/bin/awk -f
```

```
BEGIN {  
    print "\t"  
    FS = ";"  
}  
  
{  
    count=0  
    for (i=1;i<=NF;i++) { if ($i=="Luigi") { count++ } }  
    total = total + count  
}  
  
END { print "Nel file ",FILENAME, " il nome Luigi ricorre ", total, " volte \t\n" }
```

Quante volte ricorre una determinata parola in un testo (sol.)

❖ esecuzione `es.awk`

```
roberto:~$ chmod +x es.awk
```

```
roberto:~$ ./es.awk tabella.txt
```

Capovolgere un elenco

- ❖ Dato il file di testo **tabella.txt** contenete il seguente elenco (ordinato alfabeticamente):

Alberto

Barbara

Carlo

Daniela

Emma

Franco

Gino

Luca

Marco

Nino

- ❖ scrivere un file programma che eseguito da AWK sul file capovolga l'elenco (ordinando in modo inverso)

Capovolgere un elenco (sol.)

- ❖ Soluzione con comando Linux

```
sort -r tabella.txt
```

Capovolgere un elenco (sol.)

- ❖ Soluzione con file programma AWK (es.awk)

```
#!/usr/bin/awk -f
```

```
BEGIN { print "\n\tELENCO INVERTITO\n" }
```

```
{ riga[NR] = $0 }
```

```
END {
```

```
    i = NR
```

```
    while (i > 0) {
```

```
        print "\t" riga[i]
```

```
        i = i - 1
```

```
    }
```

```
    print "\n"
```

```
}
```

Capovolgere un elenco (sol.)

❖ esecuzione `es.awk`

```
roberto:~$ chmod +x es.awk
```

```
roberto:~$ ./es.awk tabella.txt
```

ELENCO INVERTITO

Nino

Marco

Luca

Gino

Franco

Emma

Daniela

Carlo

Barbara

Alberto

Classifica campionato di calcio

- ❖ Dato il file di testo **tabella.txt** contenete i risultati di calcio

```
fiorentina udinese      2 1 #prima giornata
juventus parma          2 0
milan sampdoria         0 1
atalanta lazio          0 1
chievo bologna          2 0
genoa cagliari          2 0
palermo napoli          0 3
pescara inter           0 3
roma catania            2 2
siena torino            0 0
torino pescara          3 0 #seconda giornata
bologna milan           1 3
udinese juventus        1 4
cagliari atalanta     1 1
catania genoa           3 2
inter roma              1 3
lazio palermo           3 0
napoli fiorentina       2 1
parma chievo            2 0
sampdoria siena         2 1
palermo cagliari        1 1 #Terza giornata
milan atalanta        0 1
chievo lazio            1 3
fiorentina catania      2 0
genoa juventus          1 3
napoli parma            3 1
pescara sampdoria       2 3
roma bologna            2 3
siena udinese           2 2
torino inter            0 2
parma fiorentina        1 1 #Quarta giornata
juventus chievo         2 0
sampdoria torino        1 1
atalanta palermo        1 0
bologna pescara         1 1
cagliari roma           0 3
catania napoli          0 0
inter siena             0 2
udinese milan           2 1
lazio genoa             0 1
```

Classifica campionato di calcio

- ❖ Costruire un file programma in AWK per calcolare la classifica di campionato di calcio

```
.....
SQUADRE          PUNTI
juventus          12
napoli            10
sampdoria         10
lazio             9
atalanta          7
fiorentina        7
roma              7
genoa             6
inter             6
catania           5
siena             5
torino            5
bologna           4
parma             4
udinese           4
chievo            3
milan             3
cagliari          2
palermo           1
pescara           1
.....
```

Classifica campionato di calcio (sol.)

- ❖ Soluzione con file programma AWK (`classifica.awk`)

```
#!/usr/bin/awk -f
```

```
BEGIN {
```

```
    print "\n....."
```

```
    FORMAT="\t%-18s%s\n"
```

```
    printf FORMAT,"SQUADRE","PUNTI"
```

```
}
```

Classifica campionato di calcio (sol.)

```
{
```

```
#In caso di vittoria in casa, assegno 3 punti alla prima
```

```
# 0 punti alla seconda squadra.
```

```
if ($3 > $4 ) {
```

```
    c[$1]+=3; c[$2]+=0
```

```
}
```

Classifica campionato di calcio (sol.)

#In caso di vittoria fuori casa, assegno 3 punti alla seconda

#0 punti alla prima squadra.

else if (\$3 < \$4){

 c[\$1]+=0; c[\$2]+=3

}

Classifica campionato di calcio (sol.)

#In caso di pareggio, assegno un punto a squadra.

```
else if ($3==$4){
```

```
    c[$1]+=1; c[$2]+=1;
```

```
}
```

```
c[$1] += $2
```

```
}
```

Classifica campionato di calcio (sol.)

```
END {
```

```
    ordina = "sort -k 2nr"
```

```
    for (i in c) {
```

```
        printf FORMAT,i,c[i] | ordina
```

```
    }
```

```
    close(ordina)
```

```
    print ".....\n"
```

```
}
```

Classifica campionato di calcio (sol.)

❖ esecuzione `classifica.awk`

```
roberto:~$ chmod +x classifica.awk
```

```
roberto:~$ ./classifica.awk tabella.txt
```



SQUADRE	PUNTI
juventus	12
napoli	10
sampdoria	10
lazio	9
atalanta	7
fiorentina	7
roma	7
genoa	6
inter	6
catania	5
siena	5
torino	5
bologna	4
parma	4
udinese	4
chievo	3
milan	3
cagliari	2
palermo	1
pescara	1

Lista della spesa

- ❖ Dato il file di testo **tabella.txt** contenete la lista della spesa

```
25 euro di gamberetti
15 euro di vino
5 euro di pasta
20 euro di trote
8 euro di insalata
25 euro di dolci
15 euro di tabacco
5 euro di nutella
25 euro un asciugacapelli
10 euro di pecorino
3 euro di stuzzicadenti
4 euro di tovaglioli
```

Lista della spesa

- ❖ Costruire un file programma in AWK per calcolare il totale e il peso percentuale di ogni voce di spesa.

```
.....START.....  
  
TOTALE          PRODOTTI          COSTI %  
  
160 euro  
    insalata          5%  
    vino              9.375%  
    asciugacapelli    15.625%  
    tovaglioli        2.5%  
    trote              12.5%  
    stuzzicadenti      1.875%  
    tabacco            9.375%  
    dolci              15.625%  
    pasta              3.125%  
    pecorino           6.25%  
    gambaretti         15.625%  
    nutella            3.125%  
  
.....END.....
```

Lista della spesa (sol.)

- ❖ Soluzione con file programma AWK (spesa.awk)

```
#!/usr/bin/awk -f
```

```
BEGIN {
```

```
    print "\n\t.....START.....\n"
```

```
    FORMAT="\t%-14s%-18s%s\n"
```

```
    printf FORMAT,"TOTALE","PRODOTTI","COSTI %"
```

```
}
```

Lista della spesa (sol.)

```
{  
    spesa[$4]= $1  
    somma += $1;  
}  
  
END {  
    print "\n\t",somma " euro"  
    for (i in spesa) {  
        printf FORMAT,sum,i,(100*spesa[i]) / somma "%"  
    }  
    print "\n\t.....END.....\n"  
}
```

Lista della spesa (sol.)

❖ esecuzione `spesa.awk`

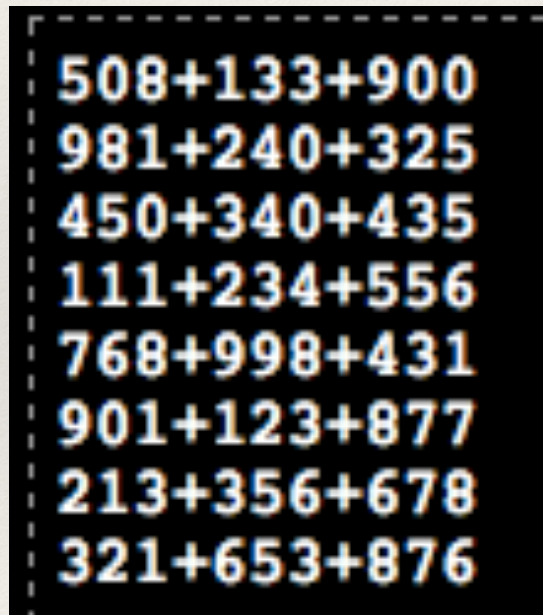
```
roberto:~$ chmod +x spesa.awk
```

```
roberto:~$ ./spesa.awk tabella.txt
```

```
.....START.....
TOTALE          PRODOTTI          COSTI %
160 euro
    insalata          5%
    vino              9.375%
    asciugacapelli    15.625%
    tovaglioli        2.5%
    trote             12.5%
    stuzzicadenti     1.875%
    tabacco           9.375%
    dolci             15.625%
    pasta             3.125%
    pecorino          6.25%
    gambaretti        15.625%
    nutella           3.125%
.....END.....
```

Sommare i numeri in colonne e righe

- ❖ Dato il file di testo **tabella.txt** contenete un elenco di numeri da sommare come segue:



```
508+133+900
981+240+325
450+340+435
111+234+556
768+998+431
901+123+877
213+356+678
321+653+876
```


Sommare i numeri in colonne e righe

- ❖ Sommare i numeri di ciascuna colonna e ciascuna riga e ottenerne il totale per colonne e per righe.

SOMMA			
508	133	900	1541
981	240	325	1546
450	340	435	1225
111	234	556	901
768	998	431	2197
901	123	877	1901
213	356	678	1247
321	653	876	1850
<hr/>			
4253	3077	5078	--> 12408 (Totale)

Sommare i numeri in colonne e righe (sol.)

- ❖ Soluzione con file programma AWK (`somma.awk`)

```
#!/usr/bin/awk -f
```

```
BEGIN {
```

```
    print "\n\t\t_____SOMMA_____ \n";
```

```
    FS="+"; OFS = "\t";
```

```
}
```

Sommare i numeri in colonne e righe (sol.)

```
{  
    a += $1; b += $2; c += $3;  
  
    tot = ($1+$2+$3);  
  
    print "\t" $1, $2, $3, " | " tot;  
}
```

Sommare i numeri in colonne e righe (sol.)

```
END {
```

```
    print "\t_____ \n";
```

```
    print "\t" a, b, c, "--> "(a+b+c)" (Totale)", "\n";
```

```
}
```


Sommare i numeri in colonne e righe (sol.)

❖ esecuzione `somma.awk`

```
roberto:~$ chmod +x somma.awk
```

```
roberto:~$ ./somma.awk tabella.txt
```

	SOMMA			
508	133	900		1541
981	240	325		1546
450	340	435		1225
111	234	556		901
768	998	431		2197
901	123	877		1901
213	356	678		1247
321	653	876		1850
<hr/>				
4253	3077	5078	-->	12408 (Totale)