

Part I – Covid Cases and Deaths

The file `/home/mathcs/courses/cs246/covid.csv` contains data about covid cases and deaths by state. Here is the beginning of the file:

```
California,12668391,107517,39512223
Colorado,1865984,15379,5758736
Connecticut,983652,12354,3565287
Delaware,348931,3584,973764
```

It is a CSV file which means that the fields are separated by commas.

Write an AWK program called `covid.awk` that prints, for each state, the number of cases per 1,000,000 population, the number of deaths per 1000 cases, and the number of deaths per 1,000,000 population. In addition, print the overall averages for all the states.

The output should look (exactly) like this:

State	Cases Per 1M Pop	Deaths Per 1K Cases	Deaths Per 1M Pop
Alabama	338542	13	4311
Alaska	412159	5	2030
Arizona	356586	13	4640
Arkansas	352112	12	4389
...			
West Virginia	389665	12	4602
Wisconsin	351028	8	2878
Wyoming	337268	11	3625
Average	328895	11	3543

To make your program work without specifying the field separator on the command line, set it in the BEGIN block with

```
FS = ","
```

Use `printf` to line up the columns and get the correct width.

If you copy the data file to your `cs246` directory, you can run the program with

```
awk -f covid.awk covid.csv
```

for testing.

Note: You need to print the header in a BEGIN block and the averages in an END block.

Part II – More Discworld

The file `/home/mathcs/courses/cs246/books` contains a listing of all the discworld characters and books, in the following format:

```
witches Witches Abroad
bursar Lords and Ladies
reg Feet of Clay
pthagonal Small Gods
gaspode Hogfather
lutze Small Gods
dean The Last Continent
dean Eric
gaspode Feet of Clay
hughnon Reaper Man
slant Jingo
cmot Soul Music
gaspode The Truth
```

The first word on each is the character(s) and the remaining words are the title of a book in which they appear. The entries are in a random order and there are multiple entries for each character.

1. Write an AWK program called `books.awk` that prints each character names followed by the list of books in which the character appears. The output should look like this:

```
carrot
  Guards! Guards!
  Jingo
  Men At Arms
  Thud!
  The Last Hero
  Feet of Clay
  Night Watch
  The Truth
  The Fifth Elephant
you-bastard
  Pyramids
esme
  Carpe Jugulum
  Lords and Ladies
  Equal Rites
  Maskerade
  Witches Abroad
  Wyrd Sisters
  The Wee Free Men
  A Hat Full Of Sky
...
```

Hints:

You can get the book name stored in a variable in several ways:

- Remember that `$0` is the entire input line. Use the `index` function to get the position of the first space and the `substr` function to extract everything after the first space.
- Use `gsub`, `gensub`, or `sub` to remove the character name from the input line (after storing the character name in a variable).

Use an array indexed by the character name to store the list of books as a single string, including newlines and spaces before each one. The array entry will accumulate the string containing the books.

2. Write an `awk` program called `characters.awk` that prints the books, followed by a list of characters appearing in the book. The output will look like this:

```
Moving Pictures
  stibbons
  nobby
  whitlow
  cmot
  gaspode
  bursar
  dean
  detritus
  poons
  librarian
  ridicully
  things
  vetinari
Thief of Time
  susan
  death
  gytha
  auditors
  war
  lutze
...
```

Hint: This is almost the same as `books.awk`. You just need to reverse the roles of the character and book.

3. For extra credit: Write a script called `books2.awk` that is the same as `books.awk` except that the list of characters is sorted. Even more extra credit if you can also sort the list of books for each character.

Hint: Use the `asorti` function. To sort the books also, instead of storing the books as a string, store them in an array. So now you will have an array of arrays.