

## 1 Count unique lines

Write an awk script that counts how many there are of each unique line of input. For example, given the input:

```
one
two and a half and more
and four
one
and four
four
```

the script should produce the output

```
2 one
1 two and a half and more
2 and four
1 four
```

the order of the output is not important. In awk, \$0 holds the current line of input. Naturally, the script should work on any normal text input. The script should accept input on stdin, and produce output on stdout. You may only use awk - no other shell commands.

Turn in your solution in a file called count\_lines.sh. Your solution to take this form:

```
awk '{/* solution program here */}'
```

That is, you are writing a bash script which contains only a single awk command.

**Hint:** you'll want to use an associative array / dictionary of counts, and print the contents of the dictionary in the END block using a for loop.

## 2 Count unique words

Write a second awk script that counts unique words instead of lines. Here, with the same input, the script should instead produce the output

```
1 a
4 and
3 four
1 half
1 more
2 one
1 two
```

again, the order of the output is not important. The individual columns/words in a line can be referenced using either numeric constants, such as `$3`, or using a variable, in which case it is `$var`. Here, `var` contains the number, and the `$` prefix indicates that you want the column with the number stored in `var`.

Turn in your solution in a file called `count_words.sh`. Again, this file should contain a single awk command.

**Hint:** you'll want to use a for loop to iterate over all the words in each line as it comes in. The variable `NF` contains the number of columns on the current line.