Language Assignment #5: Awk

Issued: Thursday, November 14
Due: Tuesday, December 3

Purpose

This assignment allows you to program in a language introduced in lecture: Awk. It was developed by, and named after, Al Aho, Peter Weinberger, and Brian Kernighan, from Bell Labs, in 1977. The GNU distribution of Awk is called Gawk, which has, since 1994, been actively maintained by Arnold Robbins.

Why Awk?

Awk is a procedural and imperative language, but based on a stream-editor model of computation. Awk is part of every Unix distribution, but has been ported to many other operating systems. Its predecessors include `sh`, `ed`, `ex`, `grep`, `egrep`, `sed`, `vi`, C, and Snobol. Its successors include Perl, Tcl, Python, and many Awk dialects.

Documentation

Awk lecture slides are at:

```
pub/slides/slides-awk.pdf
```

Awk is demonstrated by:

```
pub/sum/awk
```

Awk is also briefly described in Section 14.2.2 of our textbook.

Links to programming-language documentation can be found at:

```
http://csweb.boisestate.edu/~buff/pl.html
```

Assignment

Suppose you work for a realtor (my condolences) and your employer wants to put Ada County building-permit information on the company web page.

The following filename extensions are relevant:

`.xlsx`	Microsoft Excel Spreadsheet
`.html`	HyperText Markup Language
`.csv`	Comma-Separated Values

The building-permit data is public, but, of course, only as a `.xlsx` file. You want to process the data, eventually producing a `.html` file. You decide to use the LibreOffice program `unoconv` to batch-convert the `.xlsx` file to a `.csv` file, and then process it with an Awk script to produce the `.html` file.

Ada County provides the `.xlsx` file, from:

```
https://adacounty.id.gov/Development-Services/
      Building-Division/
```

I provide the corresponding `.csv` file, at:

```
pub/la5
```

You need to write the Awk script to produce a simple `.html` file (see below). You can view your result with a web browser (e.g., Firefox).

Hints and Advice

- You are required to use patterns/actions. Do not use an explicit loop to read input lines.
- Your program should read from `stdin` and write to `stdout`. Use no other files.

- Do not overlook Section 4.7 of (Edition 4) the Gawk manual (Defining Fields By Content). It contains the ominous "The most notorious such case is so-called 'comma separated value' (CSV) data."
- Keep your HTML simple. There is a sample skeleton at:

  ```
  pub/la5
  ```

  Simple headings are nice, but don't get carried away.
- Your employer only cares about single-family dwellings, but watch out for scruffy human-generated data. Case conversion and regular expressions can help. Furthermore, only date, subdivision name, lot, block, and value are important.