G51UST – Unix and Software Tools
Coursework 1: Sed and Awk
Weight: 40% of coursework mark (20% of total mark)
Deadline: 11/11/2013, 4pm

Preface

This coursework focuses on using the sed and awk programs. The scripts you have to produce in this coursework are some of the pieces that you will/should use for the second coursework.

Submission

You should be submitting a "tarball" cw1-<school username>.tar.gz containing the following files via **Moodle**:

- areacode.sed
- count-by-week-and-month.awk
- extractCoords.sed
- nearby-accidents.awk
- normalise.sed

Marking criteria

The next sections detail the weight that each part of the coursework will have in the mark. There is no single correct solution any of these scripts. Any script that produces the correct answer will get pass marks. To achieve higher grades, you should make appropriate use of the regular expressions, sed and awk functionalities, to create the scripts that are as compact (but intelligible) as possible. Commenting your code and using meaningful variable names (whenever appropriate) will be a bonus.

Preliminaries

In this coursework you will be using two *Open Data* datasets that have been fetched from the UK open data repository at data.gov.uk. The two datasets are:
- **Code Point Open**: dataset from Ordnance Survey that converts post codes into coordinates. Available at http://www.cs.nott.ac.uk/~jqb/codepoint.zip
- **Road Safety**: dataset from the Department of Transport, detailing car accidents for 2012 in England and Wales. Available at http://www.cs.nott.ac.uk/~jqb/accidents.zip

Moreover, you will be dealing with UK postcodes and geographical coordinates, so here is a bit of background knowledge on both:
http://en.wikipedia.org/wiki/Postcodes_in_the_United_Kingdom
http://en.wikipedia.org/wiki/Easting_and_northing
http://en.wikipedia.org/wiki/Geographic_coordinate_system

Playing with postcodes (20%)

1. Write a sed script that takes a postcode as input and filters out any non-alphanumerical character and converts lower-case characters into upcase ones:

```
$ echo "(NG8 1bb)" | sed -f normalise.sed
NG81BB
```

2. Write a sed script that takes a postcode and returns its area code in lower-case characters:

```
$ echo "NG8 1bb" | sed -f areacode.sed
ng
```

Extracting coordinates (20%)

The Code Point coordinate files use a file format called XML that, to the human eye, looks very cluttered:

```
$ head -4 codepoint/ng_position.nt
<http://data.ordnancesurvey.co.uk/id/postcodeunit/NG11AA>
<http://www.w3.org/2003/01/geo/wgs84_pos#lat>
"52.955028"^^<http://www.w3.org/2001/XMLSchema#decimal>.
<http://data.ordnancesurvey.co.uk/id/postcodeunit/NG11AA>
<http://www.w3.org/2003/01/geo/wgs84_pos#long> "-
1.141043"^^<http://www.w3.org/2001/XMLSchema#decimal>.
<http://data.ordnancesurvey.co.uk/id/postcodeunit/NG11AA>
<http://data.ordnancesurvey.co.uk/ontology/spatialrelations/easting>
"457803.00"^^<http://www.w3.org/2001/XMLSchema#decimal>.
<http://data.ordnancesurvey.co.uk/id/postcodeunit/NG11AA>
<http://data.ordnancesurvey.co.uk/ontology/spatialrelations/northing>
"340082.00"^^<http://www.w3.org/2001/XMLSchema#decimal>.
```

Write a sed script that filters all the XML clutter from the Code Point coordinate files to produce a clean output such as this:

```
$ cat codepoint/ng_position.nt | sed -n -r -f
extractCoords.sed | head -4
NG11AA lat 52.955028
NG11AA long -1.141043
NG11AA easting 457803.00
NG11AA northing 340082.00
```

Finding nearby accidents (30%)

Write an awk script that given a set of coordinates (as easting and northing) and a radius, finds the accidents that took place within the radius of the given coordinates, and from each accident it prints:

1. The distance to the coordinates
2. The coordinates of the accident (both in easting/northing and in longitude/latitude)
3. The date of the accident

4. The day of the week of the accident (number from 1 to 7)

```
$ cat DfTRoadSafety_Accidents_2012.csv | awk -v
EASTING=454590.00 -v NORTHING=339970.00 -v RADIUS=200 -f
nearby-accidents.awk
167.598 454707 340090 -1.187115 52.955382 19/04/2012 5
108.462 454548 340070 -1.189485 52.955219 10/07/2012 3
164.77 454705 340088 -1.187145 52.955364 24/12/2012 2
```

Counting nearby accidents (30%)

Write an awk script that given a list of nearby accidents as produced by `nearby-accidents.awk`, it counts the accidents that happened in a specific month (numbered from 1 to 12) and day of the week (numbered from 1 to 7):

```
$ cat nearby-accidents.txt | awk -v DAY=5 -v MONTH=09 -f
count-by-week-and-month.awk
3
```