# Task 1:

Using the Shell and a basic C program, my steps were this (in no particular order).

## Source File:

wlist\_match1.txt

## Filter Instructions:

1: Make sure all letters are lowercase

2: Remove most special symbols (will retain -,&`’), although we were allowed to retain most special symbols, I chose to remove anything that would impact how the word would be read, this includes removing preceding ‘-’, and removing ‘.’ entirely (we don’t use dots outside of punctuation of sentences in normal writing)

3: Remove all words less than 3 or greater than 15

4: Sort

5: Remove Duplicates

## Coreutils

Commands used to filter and sort data in terminal using coreutils:

* Had to use *sudo* in WSL
* *\w{3,15}* to get words of length 3 to 15
* \b[a-zA-Z&'`-]+[a-zA-Z-&'`]\*\b for regex
* *sort – u* to remove duplicates
* *tr [:upper:] [:lower:]* to make sure everything is lower case
* *shuf > Task1.sh* to shuffle and output to a file

Word Count is 1393843 using coreutils.

## Task1:

- To quickly get rid of duplicates, I sorted the array from a to z and then then just compared each item in the array to the next

- I then reversed the printing of the array so that it was from z to a, to allow for sorting a-z in the next task.

## Performance:

Time taken:

CPU cycles?

Current Notes:

**Report:**

Initially I used a for loop comparing the conditions that I wanted (word > Minimum\_Word.length), however I ended up using a Regex expression,

Initially tried removing duplicates after writing the file by iterating, however I realised I could just sort prior to iterating and this led to a performance increase from n^2 to 3n

#### Things to do:

Implement parameters for Task1  
Investigate how to improve regex for Task1 (If you can’t do it just remove it from the report)

Start report skeleton

Perf log For Shell

Create Clock and time how long Task1.c takes

Log every single action the program takes

Create an alarm function like how you saw in the lecture to make sure that the program exits gracefully

Task2  
Task3  
Task4  
Task5