

MAU11601 Assignment 2
Due 2026-10-15

1. Spreadsheet programmes tend to have weird data formats which are hard to parse but they usually have an export as csv option, which is relatively simple to parse. Csv stands for comma separated values. Each row of the table is one line of the csv file and the different columns of the row are separated by commas. In reality things are a bit more complicated because data in a cell of the table might have commas or newline characters in it so some sort of quoting mechanism is needed.

Suppose you need the data in the spreadsheet `example.xlsx` for a programme you are writing. Your programme needs the data represented as a list of lists. You've taken the first step of exporting the data as csv, so you have a csv file `example.csv`. Luckily the original table doesn't have any cells with commas or newlines, so there is no quoting to worry about. How should you create your list of lists from the csv file?

2. One of the earliest sort algorithms is quicksort. You take the first item in the list, split the rest of the items into two lists, the ones which should come before it and the ones which should come after it, sort those, and then combine the results. Here's an implementation.

```
(define (qsort l)
  (if (empty? l)
      l
      (append (qsort (filter (curry >= (first l)) (rest l)))
              (list (first l))
              (qsort (filter (curry < (first l)) (rest l))))))
```

For simplicity I've assumed these are list of numbers, so the order relation is the usual `<` function.

- (a) Is `qsort` tail recursive?
- (b) How do we know `qsort` terminates?
- (c) Suppose we replace `>=` with `>`. What goes wrong?
- (d) Suppose we replace `<` with `<=`. What goes wrong?
- (e) What happens if the list was already in sorted order? How many comparisons are made?