COMP9021, Session 1, 2014

The aim of this lab is to practice the use of arrays.

1 ☞ Computing statistics on the characters in a text

Write a program that takes as input a sequence of characters all on one line, ending in carriage return (anything input after the first carriage return will be ignored), and that outputs how many times each digit has been entered, provided it has been entered at least once, following this kind of interaction:

```
$ a.out
Enter characters: some input with no digits
Digits:
Count:
$ a.out
Enter characters: Some input with digits such as 23 and 103
Digits:    0   1   2   3
Count:     1   1   1   2
$ a.out
Enter characters: 000112    0129999 010
Digits:    0   1   2   9
Count:     6   4   2   4
$ a.out
Enter characters: 2 qw12e 12 e65 3 wq
Digits:    1   2   3   5   6
Count:     2   3   1   1   1
```

2 Magic squares

A 3×3 square whose cells contain every digit in the range 1–9 is said to be *magic* of the sums of the rows, the sums of the columns and the sums of the diagonals are all equal numbers.

Write a program that generates all magic squares.

Here is the beginning of a possible run of the program ("possible", as the order of the output solutions can vary):

```
$ a.out
   4  9  2
   3  5  7
   8  1  6

   6  7  2
   1  5  9
   8  3  4

   ...
```

3 ☞ Counting the number of lines in a grid

Consider a grid of size 13 x 13 delimited with # symbols. Consecutive # symbols are separated by a space in the horizontal dimension, but not in the vertical dimension. The grid possibly contains a number of * symbols at the intersection of horizontal and vertical lines that go through # symbols, and no other symbols anywhere else. There might be any amount of space outside the grid, on all sides. Here is an example of a such a grid, stored in a file grid_sample.txt that you can download.

```
# # # # # # # # # # # # #
# *     *   * * *   *   #
#   *   * *   * *       #
# *     * *     *       #
#         * * * *   * * #
#     * *     * *     * #
# *     * *     *       #
#       *       * *     #
#                       #
#       *       *       #
# * *   * *     *       #
#     * *     * *     * #
# # # # # # # # # # # # #
```

Write a C program that outputs how many horizontal, vertical and diagonal lines made up of (at least 2) * symbols are contained in the grid. With the sample grid above, the program would be run as follows:

```
$ a.out <sample_grid.txt
The grid contains
  - 14 horizontal lines
  - 9 vertical lines
  - 22 diagonal lines
```