FIT5216: Modelling Discrete Optimization Problems
Inclass Task 1: Cryptarithms

1 Problem Statement

A *cryptarithm* is a mathematical puzzle which requires determining the digit for each letter in an equation. The most famous cryptarithm is SEND + MORE = MONEY. That is we need to determine which digit each the letters represent so that,

$$\begin{array}{cccccc} & & \mathrm{S} & \mathrm{E} & \mathrm{N} & \mathrm{D} \\ + & & \mathrm{M} & \mathrm{O} & \mathrm{R} & \mathrm{E} \\ \hline = & \mathrm{M} & \mathrm{O} & \mathrm{N} & \mathrm{E} & \mathrm{Y} \end{array}$$

The rules of cryptarithms are:

- Each letter represents a different digit
- The first letter in each word cannot be 0 (otherwise it would not be a proper number)
- The arithmetic equation must hold.

This project will require modelling and solving crytarithm problems.

A MiniZinc model for this problem would be

```
var 1..9: S;
var 0..9: E;
var 0..9: N;
var 0..9: D;
var 1..9: M;
var 0..9: O;
var 0..9: R;
var 0..9: Y;

constraint                1000 * S + 100 * E + 10 * N + D
            +             1000 * M + 100 * O + 10 * R + E
            = 10000 * M + 1000 * O + 100 * N + 10 * E + Y;

include "alldifferent.mzn";
constraint alldifferent([S,E,N,D,M,O,R,Y]);

solve satisfy;
```

Part 1 - SEND+MORE=MONEY

Simply submit the provided `smm.mzn` model. This will test that MiniZinc is installed and working correctly.

Part 2 - SNAKE+SNAKE=RATTLE

Build a MiniZinc model `snake.mzn` which solves the problem,

$$\begin{array}{ccccccc} & & S & N & A & K & E \\ + & & S & N & A & K & E \\ \hline = & R & A & T & T & L & E \end{array}$$

You should determine at least one solution.

Part 3 - SEND+MOST=MONEY

Build a MiniZinc model `most.mzn` which solves the problem,

$$\begin{array}{cccccc} & & S & E & N & D \\ + & & M & O & S & T \\ \hline = & M & O & N & E & Y \end{array}$$

and maximizes the value of the word MONEY. You will need to add an *objective function* of the form

`solve maximize` f

where f is the expression to be maximized.

Part 4 - Attemptation

Attemptation is a more general form of cryptarithm problem where we are given a partially filled codex, showing the set of possible values. Some values may be used more than once. For example

$$\begin{array}{ccccccc} & P & L & E & A & S & E \\ + & & S & O & L & V & E \\ + & P & U & Z & Z & L & E \\ \hline = & Q & Q & Q & Q & Q & Q \end{array}$$

with codex

0	1	2	3	5	6	7	7	8	9
						Z	Q		

means that Z and Q are both the digit 7, and the remaining letters take different values in the codex.

Build a MiniZinc model `attempt.mzn` to solve the above attemptation puzzle.

2 Instructions

Edit the provided `mzn` model files to solve the problems described above. Your implementations can be tested locally by using the *Run* icon in the MiniZinc IDE or by using,

```
minizinc ./modelname.mzn
```

at the command line.