Modelling for Combinatorial Optimisation (1DL451) and Constraint Programming (1DL442) Uppsala University – Autumn 2023 MiniZinc Exercises (optional): Solutions

Prepared by Gustav Björdal and Pierre Flener

Version of 2nd August 2023

Listing 1: Solution for Ex1.mzn

```
int: n = 5;
int: m = 10;
array[1..m] of int: A = [1,2,1,3,1,4,1,5,1,6];

array[1..m*n] of int: B = [ a | a in A, i in 1..n];

output [show(B)];
```

Listing 2: Solution for Ex2.mzn

```
int: n = 5;
int: m = 6;
array[1..m] of int: A = [1,2,3,4,5,6];

array[1..m*n] of int: B = [ a | i in 1..n, a in A];

output [show(B)];
```

Listing 3: Solution for Ex3.mzn

```
int: n = 4;
int: m = 3;
array[1..n, 1..m] of int: A = [|1,2,3|4,5,6|7,8,9|5,5,5|];

% array[1..n] of int: B = [sum(j in 1..m)(A[i,j] = 5) | i in 1..n];
%% or, more elegantly:
array[1..n] of int: B = [count(A[i,..],5) | i in 1..n];

output [show(B)];
```

Listing 4: Solution for Ex4.mzn

```
include "globals.mzn";

int: n = 4;

array[1..n] of var 1..n*n: X;
array[int] of var int: Y = [abs(X[i]-X[j]) | i,j in 1..n where i<j];

constraint all_different(X);
constraint all_different(Y);

output [show(X), "\n", show(Y)];
```

Listing 5: Solution for Ex5.mzn

```
include "globals.mzn";
int: n = 3;
array[1..n] of int: A = [2,1,3];
array[1..n] of int: B = [9,7,5];

array[1..n] of int: C = [B[i] | i in arg_sort(A)];

output [show(C)];
```

Listing 6: Solution for Ex6.mzn

```
int: n = 4;

array[int,int] of int: A = array2d(1..(n*(n-1) div 2), 1..3,
  [[f,s,f*n+s][i] | f,s in 1..n where f<s, i in 1..3 ]);

output [show2d(A)];
```