NOPT042 Constraint programming: Tutorial 10 - Modeling with sets

```
In [1]: %load_ext ipicat
```

```
Picat version 3.5#5
```

Modelling with sets

In Picat, the cp solver doesn't work natively with sets and set constraints (unlike e.g. MiniZinc). Instead, we can model a set as an array (or a list) representing its characteristic vector. For a collection of sets, we can use a matrix or a list of lists.

- A subset $S \subseteq \{1, \ldots, n\}$: `S = new_array(N), S :: 0..1`
- Fixed cardinality subset: `exactly(K, S, 1)`
- Bounded cardinality subset: `at_most(K, S, 1)`, `at_least(K, S, 1)` (or we could use `sum`)

Set operations can be computed using bitwise logical constraints, e.g.

```
SintersectT = [S[I] #/\ T[I] : I in 1..N]
```

Alternatively, we could use a list of elements and require that the list is strictly increasing. In that case, we need to declare a list of length N and have a decision variable for the length of the list. We can use 0 as a dummy value denoting that there are no more elements

```
S = new_list(N),
S :: 0..N,
SizeOfS :: 0..N,
increasing_strict(S[I] : I in 1..SizeOfS),
foreach(I in SizeOfS+1..N)
    S[I] #= 0
end
```

A partition of $\{1, \ldots, n\}$ with k classes can be modelled as a function $\{1, \ldots, n\} \rightarrow \{1, \ldots, k\}$:

```
Partition = new_array(N),
Partition :: 1..K
```

Do not forget about symmetry breaking, e.g. `Partition[1] #= 1` or

```
    foreach(I in 1..K)
        Partition[I] #<= I
    end.
```

Similarly for a collection of k pairwise disjoint subsets---using 0 to denote that an element is not covered by any subset.

Example: Finite projective plane

> A projective plane geometry is a nonempty set X (whose elements are called "points"), along with a nonempty collection L of subsets of X (whose elements are called "lines"), such that:
>
> - For every two distinct points, there is exactly one line that contains both points.
> - The intersection of any two distinct lines contains exactly one point.
> - There exists a set of four points, no three of which belong to the same line.

(from Wikipedia)

A projective plane of **order** `N` has $M = N^2 + N + 1$ points and the same number of lines, each line must have $K = N + 1$ points and each point must lie on K lines. A famous example is the Fano plane where $N = 2$, $M = 7$, and $K = 3$.

If the order N is a power of a prime power, it is easy to construct a projective plane of order N. It is conjectured otherwise, no projective plane exists. For $N = 10$ this was famously proved by a computer-assisted proof (that finished in 1989). The case $N = 12$ remains open.

Example: Ramsay's partition

Partition the integers 1 to n into three parts, such that for no part are there three different numbers with two adding to the third. For which n is it possible?

Homework: golfers

The Social Golfer Problem, see also problem description on CSPLib.

There are $n = g \times s$ golfers who play golf once per week. Each week they play in g groups of s golfers per group. Create a schedule for w weeks such that **no**

golfer plays in the same group as any other golfer on more than one occasion, i.e., maximum socialization. If it is possible, output `yes` and some reasonable representation of the schedule.

An instance is given by the triple of parameters (g, s, w). Running

```
picat golfers.pi 3 2 5
```

should output `yes` and a valid schedule in some reasonable representation, for example:

```
yes
[1,1,1,1,1]
[1,2,2,2,2]
[2,1,2,3,3]
[2,3,3,1,2]
[3,2,3,3,1]
[3,3,1,2,3]
```

where each row represents a schedule for one golfer (there are 6 golfers), the numbers are the groups. The output of

```
picat golfers.pi 2 2 4
```

should include `failed` (it can be in stderr as Picat normally does).

(This is a hard problem, so don't be surprised if your model won't be able to solve even relatively small instances. The instanc e `8 4 10` was a somewhat well-known open problem, solved only in 1996.)