Simple Modelling II
Iteration

CLP languages do not provide iteration directly; you must use recursion to do this for you.
Converting a while loop to a recursive call

while ( T > 1)
do
    somecode;
    make T smaller;
done.

Becomes:

my_pred(1, Othervars,...).
my_pred(T, Othervars, ...) :-
    somecode,
    makeTsmaller(T, SmallerT),
    my_pred(SmallerT, ......).
Looping through a list

Sometimes you want to loop through a list. For example the following sums a list in CLP(R).

```prolog
listsum([], 0).
listsum([H|T], H + Sum) :-
    listsum(T, Sum).
```
Optimization

Unfortunately we are forced to solve minimization (or maximization) problems by inspection in CLP(R). Nevertheless it is theoretically possible to add a minimization derivation step to the resolution model.
Suppose $G_1$ is of the form $L_1, L_2, \ldots, L_n$ where $L_1$ is $\text{minimize}(G, E)$ then there are two cases:

1. There is at least one solution $\phi$ of $\langle G \mid C_1 \rangle$ with $\phi(E) = m$, and for all other solutions $\theta$ of $\langle G \mid C_1 \rangle$, $m \leq \theta(E)$. Then $C_2$ is $C_1 \land E = m$ and $G_2$ is $G, L_2, \ldots, L_n$

2. If evaluation can detect that $\langle G \mid C_1 \rangle$ has no solutions or has an unbounded minimum then $C_2$ is false and $G_2$ is the empty goal.