Constraint Programming
Prof. Dr. Thom Frühwirth, Marc Meister

assignment #6 (winter term 2005)
solutions will be presented Tuesday, 6-Dec-2005, 2 PM, o27/2203
`http://www.informatik.uni-ulm.de/pm/index.php?id=112`

Exercise 1. (a) Implement the lecture's CCLP-example `hamming` in CHR. In order to view the computed results in the infinite sequence of hamming numbers, implement an additional, recursivly defined predicate `observe`, which watches `hamming` and outputs results as soon as they are available with (`write(...),nl`).
Note the order in the call – `observe(S),hamming(S)` – and comment on what happens when exchanging `observe(S)` and `hamming(S)`.

(b) Implement a corresponding CCLP-example `plussing/2` in CHR. `plussing(N,L)` computes the infinite sequence of numbers `L`, which can be computed arbitrary addition of the given positive integers from the list `N`.
Example: For `N=[3,7]` we have `L=[3,6,7,9,10,...]`.

Constraint Handling Rules (CHR)

Exercise 2. Compare the following CHR programs, which consist of *one* of the given rules by posing the given queries. Check your answers with the system's answers. Make sure, you understand why seemingly innocous rules produce different answers.

```
c1 @ c(X), c(X) <=> q(X,X).
c2 @ c(X), c(Y) <=> r(X,Y).
c3 @ c(X), c(X) ==> q(X,X).
c4 @ c(X), c(Y) ==> r(X,Y).
```

Queries:

a) `c(X), c(X)`
b) `c(X), c(Y)`
c) `c(X), c(Y), X=Y`

More variants:

```
q1 @ p(X,Z), q(Z,Y) <=> q(X,Y).
q2 @ q(Z,Y), p(X,Z) <=> q(X,Y).
q3 @ p(X,Z), q(Z,Y) ==> q(X,Y).
q4 @ q(Z,Y), p(X,Z) ==> q(X,Y).
q5 @ p(X,Z) \ q(Z,Y) <=> q(X,Y).
q6 @ q(Z,Y) \ p(X,Z) <=> q(X,Y).
```

Queries:

d) `p(A,B), q(B,C)`
e) `p(A,B), q(B,C), p(D,A)`

Comment on the system's answers for queries a) to e).
Comment on the system's answers for the rule `q5` and the following two queries.

- `p(X,C), p(Y,C), q(C,A)` und
- `p(Y,C), p(X,C), q(C,A)`.

Exercise 3. Implement the constraints `less/2` (encoding $<$) und `leq/2` (encoding $\leq$) and their mutual relations/interactions in CHR. You may find the lecture's CHR program for the $\leq$ constraint helpful.
For an example query, take your last name as a sequence of variables with $\leq$ constraints between succeeding characters.
The name *Fruehwirth* translates to the query
`F leq R, R leq U, U leq E, E leq H, H leq W, W leq I, I leq R, R leq T, T leq H`
with answer `F leq E, H=E, I=E, R=E, T=E, U=E, W=E`.
Tests should include (at least) three more queries consisting of combined `less` and `leq` constraints.