Practical 2a

[redacted]

Finding whether or not, a given relation is:
i.Reflexive
ii. Antisymmetric
iii. Transitive
iv. Partial order

1

```
(%i1) kill(all);
(%o0) done
```

ex 7 : Rosen

```
(%i1) A:makelist(k,  k,  1,  4);
(%o1) [1,2,3,4]

(%i2) R1:[[1, 1], [1, 2], [2, 1], [2, 2], [3, 4], [4, 1], [4, 4]];
(%o2) [[1,1],[1,2],[2,1],[2,2],[3,4],[4,1],[4,4]]
```

A relation R on a set A is called reflexive

if (a, a) ∊ R for every element a ∊ A.

```
(%i3) member(a,  [a,  b,  c]);
(%o3) true

(%i5) s:0;
      for i:1 thru length(A) do(
         if(member([A[i],  A[i]],  R1) = true) then (s:s+1)
      );
(%o4) 0
(%o5) done

(%i6) s;
(%o6) 3

(%i7) if(s=length(A)) then print("reflexive") else print("Not reflexive");
      Not reflexive
(%o7) Not reflexive
```

```
(%i8) checkReflexive(A,  R):=block(
        [s:0],
        for i:1 thru length(A) do(
            if(member([A[i],  A[i]],   R) = true) then (s:s+1)),
        if(s=length(A)) then return("reflexive") else return("Not reflexive")
      );
(%o8) checkReflexive(A,R):=block([s:0],for i thru length(A)
      do if member([A_i,A_i],R)=true then s:s+1 ,if s=length(A)
      then return(reflexive) else return(Not reflexive))

(%i9) checkReflexive(A,   R1);
(%o9) Not reflexive
```

ex 7 : Rosen

R 1 = {(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)},
R 2 = {(1, 1), (1, 2), (2, 1)},
R 3 = {(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)},
R 4 = {(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)},
R 5 = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)},
R 6 = {(3, 4)}.

```
(%i10) R2:[[1,  1],  [1,  2],   [2,  1]];
(%o10) [[1,1],[1,2],[2,1]]

(%i11) checkReflexive(A,  R2);
(%o11) Not reflexive

(%i12) R3:[[1, 1], [1, 2], [1, 4], [2, 1], [2, 2], [3, 3], [4, 1], [4, 4]];
(%o12) [[1,1],[1,2],[1,4],[2,1],[2,2],[3,3],[4,1],[4,4]]

(%i13) checkReflexive(A,  R3);
(%o13) reflexive

(%i14) R1:[[1, 1], [1, 2], [2, 1], [2, 2], [3, 4], [4, 1], [4, 4]];
(%o14) [[1,1],[1,2],[2,1],[2,2],[3,4],[4,1],[4,4]]

(%i15) R2:[[1, 1], [1, 2], [2, 1]];
(%o15) [[1,1],[1,2],[2,1]]

(%i16) R3: [[1, 1], [1, 2], [1, 4], [2, 1], [2, 2], [3, 3], [4, 1], [4, 4]];
(%o16) [[1,1],[1,2],[1,4],[2,1],[2,2],[3,3],[4,1],[4,4]]
```

```
(%i17) R4:[[2, 1], [3, 1], [3, 2], [4, 1], [4, 2], [4, 3]];
(%o17) [[2,1],[3,1],[3,2],[4,1],[4,2],[4,3]]

(%i18) R5:[[1, 1], [1, 2], [1, 3], [1, 4], [2, 2], [2, 3], [2, 4], [3, 3], [3, 4], [4, 4]];
(%o18) [[1,1],[1,2],[1,3],[1,4],[2,2],[2,3],[2,4],[3,3],[
       3,4],[4,4]]

(%i19) R6:[[3, 4]];
(%o19) [[3,4]]

(%i20) R7:[[1,  1],  [3, 4]];
(%o20) [[1,1],[3,4]]

(%i21) R8:cartesian_product_list(A,   A);
(%o21) [[1,1],[1,2],[1,3],[1,4],[2,1],[2,2],[2,3],[2,4],[
       3,1],[3,2],[3,3],[3,4],[4,1],[4,2],[4,3],[4,4]]

(%i22) checkReflexive(A,   R1);
(%o22) Not reflexive

(%i23) checkReflexive(A,   R2);
(%o23) Not reflexive

(%i24) checkReflexive(A,   R3);
(%o24) reflexive

(%i25) checkReflexive(A,   R4);
(%o25) Not reflexive

(%i26) checkReflexive(A,   R5);
(%o26) reflexive

(%i27) checkReflexive(A,   R6);
(%o27) Not reflexive

(%i28) checkReflexive(A,   R7);
(%o28) Not reflexive

(%i29) checkReflexive(A,   R8);
(%o29) reflexive
```