

```

> LexProd:= proc(G::seq(Graph), $)
uses GT= GraphTheory;
local
  LexProd2:= proc(G::Graph, H::Graph)
local
  (Vg,Vh) := op(GT:-Vertices~([G,H])),
  (ng,nh) := op(`..`~(1, nops~([Vg,Vh]))),
  (Ng,Nh) := op(op~(4, [G,H])),
  i, j, J,
  P:= [seq](seq([i,j], j= nh), i= ng),
  k:= 0, K:= table((p-> op(p)= ++k)~(P))
;
  GT:-Graph(
(P),
    (
      ((i,j)-> {
        seq(seq(K[k,J], k= Ng[i]), J= nh),
        seq(K[i,k], k= Nh[j])
      })
      @op
    )~(P)
  )
end proc
;
if nargs=0 then error "at least 1 graph needed"
elif nargs=1 then G[1]
else foldl(LexProd2, args)
fi
end proc
:

```

Error, `(` unexpected

>