

```
> #Hrudai Battini HW 6
read "/Users/hb334/Documents/M5.txt";
Help5( )
with(linalg);
```

RecToSeq(INI,REC,N), GrowthC(INI,REC,K), GrowthCe(REC)

LeslieMod(SUR,FER): e.g. LeslieMod([9/10,9/10],[0,1,1]);

LeslieMat(SUR,FER); e.g. LeslieMat([9/10,9/10],[0,1,1]);

() [*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, subbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*]

(1)

```
> #1
#N0(t) = 0.1N0(t) + 1.2N1(t) + 0.9N2(t) + 0.1N3(t).
#N1(t) = 0.95N0(t).
#N2(t) = 0.97N1(t).
#N3(t) = 0.9N2(t).
#Using Substitution: N0(t) = 0.1N0(t) + 1.2(0.95N0(t)) + 0.9(0.97(0.95N0(t))) + 0.1(0.9(0.97(0.95N0(t))))).
REC := [0.1, 1.2*0.95, 0.9*0.97*0.95, 0.1*0.9*0.97*0.95];
Digits := 8;
print(GrowthCe);
GrowthCe(REC);
```

REC := [0.1, 1.140, 0.82935, 0.082935]

Digits := 8

```
proc(REC)
  local x, i; evalf([solve(1 - add(REC[i]/x^i, i = 1..nops(REC)))])[1]
end proc
```

1.3857326

(2)

```
> #2
print(LeslieMat);
Digits;
SUR := [0.95, 0.97, 0.9];
FER := [0.1, 1.2, 0.9, 0.1];
A := LeslieMat(SUR, FER);
E := eigenvals(A)[1];
is(E = GrowthCe(REC));
```

```

proc(SUR, FER)
  local i, A;
  if not (type(SUR, list) and type(FER, list) and nops(SUR) + 1 = nops(FER)) then
    print(`bad input`); RETURN(FAIL)
  end if;
  A := nops(SUR);
  matrix([FER, seq([0$i - 1, SUR[i], 0$A + 1 - i], i=1..A)])
end proc

```

```

      8
      SUR := [0.95, 0.97, 0.9]
      FER := [0.1, 1.2, 0.9, 0.1]
      A := [
        [ 0.1  1.2  0.9  0.1 ]
        [ 0.95  0  0  0 ]
        [ 0  0.97  0  0 ]
        [ 0  0  0.9  0 ]
      ]
      E := 1.3857326
      true

```

(3)

```

> #3
P := Matrix([[0.5, 0.5/3, 0.5/3, 0.5/3], [0.2, 0.4, 0.2, 0.2], [0.7/3,
0.7/3, 0.3, 0.7/3], [0.8/3, 0.8/3, 0.8/3, 0.2]]);
P2 := evalm(P^1000);

#Around 32/100 stay on S1, 26/100 on S2, 23/100 on S3, and 19/100
on S4. The page ranks of these four websites are S1,S2,S3,S4 in
that order .

```

```

P := [
  [ 0.5  0.166667  0.166667  0.166667 ]
  [ 0.2  0.4  0.2  0.2 ]
  [ 0.233333  0.233333  0.3  0.233333 ]
  [ 0.266667  0.266667  0.266667  0.2 ]
]

```

```

P2 :=
[ [ 0.315287668082312, 0.262739873626865, 0.225205697940426,
0.197054861317212 ],
[ 0.315287217777802, 0.262739498372893, 0.225205376294033, 0.197054579876795 ],
[ 0.315286896131409, 0.262739230334078, 0.225205146546384, 0.197054378847729 ],
[ 0.315287541642178, 0.262739768260026, 0.225205607625957, 0.197054782292100 ] ]

```

(4)