```
> #OK to post
> #Anne Somalwar, 9.20.2021, hw5
>
>
> #0
> #canonical form: a(n) = -5 a(n − 2) - 6 a(n − 4)
>
> RecToSeq :=proc(INI, REC, N) local i, k, L, newguy :
  if not (type(INI, list) and type(REC, list) and nops(INI) =nops(REC) and type(N, integer)
      and N ≥ nops(INI)) then
   print(`bad innput`) :
   RETURN(FAIL) :
  fi:

  k := nops(INI) :

  L := INI:

  while nops(L) < N do
   newguy := add(REC[i]*L[-i], i=1 ..k) :
   L := [op(L), newguy] :
  od:
  L :
  end:
>
> RecToSeq([1, 2, 4, 11], [0,-5, 0, -6], 1000)
> # a(1000)=
      181801458979349684211926335397716595590116925130008115201730179162903000957
      919477420992513491076776799335003400559596244171485816127673964664251546606
      181311762839416505521709454841943997493283513047867597347184546959401904109
      745684403540309
>
>
> #1
> GrowthC :=proc(INI, REC, K) local L, a, b :
  L := RecToSeq(INI, REC, K) :

  a := L[-1]/L[-2] :
  b := L[-2]/L[-3] :

  if abs(a-b) < 1/10^(Digits + 3) then
  RETURN(evalf(a)) :
  else
  print(` make `, K, `bigger `) :
   RETURN(FAIL) :
  fi:
  end:
> GrowthC([1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1], 100)
                                                                                (1)
```

```
                                 1.999018633                                  (1)
> GrowthCe :=proc(REC) local x, i :
  evalf([solve(1-add(REC[i]/x^i, i=1..nops(REC)))])[1] :

  end:
> GrowthCe( [1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
                                 1.999018633                                  (2)
> with(LinearAlgebra) :
>
>
> #2
> LeslieMod :=proc(SUR, FER) local i, L, A :

   if not (type(SUR, list) and type(FER, list) and nops(SUR) + 1 = nops(FER)) then
    print(`bad input`) :
   RETURN(FAIL) :
   fi:

  A := nops(SUR) :

  L[0] := 1 :

  for i from 1 to A do
   L[i] := L[i-1]* SUR[i] :
  od:

  [seq(FER[i + 1]* L[i], i=0..A)] :

  end:
```

$$> \ LeslieMod\left(\left[\frac{99}{100}, \frac{99}{100}\right], \left[0, \frac{1}{2}, \frac{1}{4}\right]\right)$$

$$\left[0, \frac{99}{200}, \frac{9801}{40000}\right] \qquad (3)$$

```
> GrowthCe(%)
                                 0.8795363925                                 (4)
> LeslieMat :=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) :
   fi:

  A := nops(SUR) :

  matrix([ FER, seq([0$(i-1), SUR[i], 0$(A + 1-i)], i=1..A)]) :
  end:
```

> $LeslieMat\left(\left[\frac{99}{100}, \frac{99}{100}\right], \left[0, \frac{1}{2}, \frac{1}{4}\right]\right)$

$$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ \frac{99}{100} & 0 & 0 \\ 0 & \frac{99}{100} & 0 \end{bmatrix} \tag{5}$$

> $Eigenvalues(\%)$

$$\left[\left[\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{200} + \frac{33}{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right],\right. \tag{6}$$

$$\left[-\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{400} - \frac{33}{2\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right.$$

$$\left. + \frac{3\,\mathrm{I}\sqrt{3}\left(\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{30} - \frac{220}{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right)}{40}\right],$$

$$\left[-\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{400} - \frac{33}{2\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right.$$

$$\left.\left. - \frac{3\,\mathrm{I}\sqrt{3}\left(\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{30} - \frac{220}{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right)}{40}\right]\right]$$

> $evalf\left(\frac{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}{200} + \frac{33}{\left(980100 + 3300\sqrt{61809}\right)^{1/3}}\right)$

$$0.8795363925 \tag{7}$$

>

>

>

>

>

> #3

>

> $GrowthCe([0, 0, 0, 0.57 \cdot 0.28 + 0.16, 0.57 \cdot 0.72 + 0.41])$

```
                              1.028168279                                   (8)
> 
> 
> #4
> 
> PlantGseq :=proc(alpha, beta, gamma, sigma, INI2, K) local REC2 :
   if not (type(INI2, list) and type(K, integer) and type(alpha, float) and type(beta, float)
      and type(gamma, integer) and type(sigma, float) and K ≥ nops(INI2) and 0 ≤ alpha ≤ 1
      and  0 ≤ beta ≤ 1 and  0 ≤ sigma ≤ 1) then
   print( `bad innput`) :
   RETURN(FAIL) :
   fi:
   REC2 := [alpha · sigma · gamma, beta · sigma^2 · (1 − alpha) · gamma] :
   RecToSeq(INI2, REC2, K) :
   end:
> PlantGseq(0.5, 0.25, 2, 0.8, [100, 80], 21);
[100, 80, 80.000000, 76.80000000, 74.24000000, 71.68000000, 69.22240000, 66.84672000,   (9)
    64.55296000, 62.33784320, 60.19874816, 58.13305344, 56.13824246, 54.21188252,
    52.35162481, 50.55520105, 48.82042081, 47.14516882, 45.52740239, 43.96514892,
    42.45650352]
> 
> PlantGseq(0.6, 0.3, 2, 0.8, [100, 96], 21)
[100, 96, 107.52000, 117.9648000, 129.7612800, 142.6902221, 156.9139458, 172.5546061,   (10)
    189.7544040, 208.6686153, 229.4681472, 252.3409206, 277.4935912, 305.1534130,
    335.5702921, 369.0190446, 405.8018797, 446.2511298, 490.7322533, 539.6471367,
    593.4377253]
> 
> 
> 
> #5
> PlantGseqGC :=proc(alpha, beta, gamma, sigma) local REC :
   if not (type(alpha, float) and type(beta, float) and type(gamma, integer) and type(sigma, float)
      and 0 ≤ alpha ≤ 1 and  0 ≤ beta ≤ 1 and  0 ≤ sigma ≤ 1) then
   print( `bad innput`) :
   RETURN(FAIL) :
   fi:
   REC := [alpha · sigma · gamma, beta · sigma^2 · (1 − alpha) · gamma] :
   GrowthCe(REC) :
   end:
> 
> #An example where the population deteriorates
> PlantGseqGC(0.5, 0.5, 1, 1.0)
                              0.8090169944                                  (11)
> 
> #An example where the population is constant
> PlantGseqGC(1.0, 0.0, 1, 1.0)
                                                                            (12)
```

```
                                    1.                                    (12)
> #An example where the population explodes
> PlantGseqGC(0.75, 0.75, 6, 1.0)
                               4.737468593                                (13)
>
```