Deven Singh, Assignment 5, 09/20/2021
OK TO POST HOMEWORK

```
#Question 0
eq := {6·a(n-1)+a(n+3)+5·a(n+1)=0};
```

$$eq := \{6\,a(n-1)+a(n+3)+5\,a(n+1)=0\} \tag{1}$$

```
subs(n=n-3, eq);
```

$$\{6\,a(n-4)+a(n)+5\,a(n-2)=0\} \tag{2}$$

```
#a(n) = 0·a(n-1)-5·a(n-2)+0·a(n-3)-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:

A := RecToSeq([1, 2, 4, 11], [0,-5, 0,-6], 1000):
A[1000];
```

$$\begin{aligned}&1818014589793496842119263353977165955901169251300081152017301791629030009579195\\&4774209925134910767767993350034005595962441714858161276739646642515466061815\\&3117628394165055217094548419439974932835130478675973471845469594019041097455\\&684403540309\end{aligned} \tag{3}$$

```
evalf(%);
```

$$1.818014590\ 10^{239} \tag{4}$$

```
#Question 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([0, 1, 1, 2, 4, 8, 16, 32, 64, 128], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1], 1000);
```

1.999018633 **(5)**

```
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 **(6)**

```
#Question 2
#Part i
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([.99, .99], [0, 1/2, 1/4]);
```

[0, 0.4950000000, 0.2450250000] **(7)**

```
GrowthCe(%);
```

0.8795363925 **(8)**

```
#Part ii
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:
with(LinearAlgebra) :
LeslieMat([.99, .99], [0, 1/2, 1/4]);
```

$$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ 0.99 & 0 & 0 \\ 0 & 0.99 & 0 \end{bmatrix} \tag{9}$$

```
with(LinearAlgebra) :
E := Matrix([[0, 1/2, 1/4], [0.99, 0, 0], [0, .99, 0]]);
evalf(Eigenvalues(E));
```

$$E := \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ 0.99 & 0 & 0 \\ 0 & 0.99 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.879536392451194 + 0.\,\mathrm{I} \\ -0.439768196225596 + 0.291870175308380\,\mathrm{I} \\ -0.439768196225596 - 0.291870175308380\,\mathrm{I} \end{bmatrix} \tag{10}$$

```
#Question 3
GrowthCe([0, 0, 0, .16, .41]);
```

$$0.8879729192 \tag{11}$$

```
#Question 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:
PlantGseq := proc(alpha, beta, gamma, sigma, INI, K) local a, b :
if not(type(INI, list) and nops(INI) = 2 and type(K, integer) and K ≥ nops(INI)) then
  print(`bad input`) :
  RETURN(FAIL) :
 fi:
 a := alpha·sigma·gamma :
 b := beta·sigma^2·(1 − alpha)·gamma :
 RecToSeq(INI, [a, b], K) :
 end:
 PlantGseq(0.5, 0.25, 2.0, 0.8, [100, 80], 20);
[100, 80, 80.000000, 76.80000000, 74.24000000, 71.68000000, 69.22240000, 66.84672000,          (1)
    64.55296000, 62.33784320, 60.19874816, 58.13305344, 56.13824246, 54.21188252,
    52.35162481, 50.55520105, 48.82042081, 47.14516882, 45.52740239, 43.96514892]
 PlantGseq(0.6, 0.3, 2.0, 0.8, [100, 96], 20);
[100, 96, 107.52000, 117.9648000, 129.7612800, 142.6902221, 156.9139458, 172.5546061,          (2)
    189.7544040, 208.6686153, 229.4681472, 252.3409206, 277.4935912, 305.1534130,
    335.5702921, 369.0190446, 405.8018797, 446.2511298, 490.7322533, 539.6471367]
#Question 5
 GrowthCe := proc(REC) local x, i :
evalf([solve(1 - add(REC[i]/x^i, i = 1 ..nops(REC)))])[1] :

end:
PlantGseq2 := proc(alpha, beta, gamma, sigma) local a, b :
if not(type(alpha, float) and type(beta, float) and type(gamma, float) and type(sigma, float)) then
  print(`bad input`) :
  RETURN(FAIL) :
 fi:
 a := alpha·sigma·gamma :
 b := beta·sigma^2·(1 − alpha)·gamma :
 GrowthCe([a, b]) :
 end:
```

PlantGseq2(0.5, 0.25, 2.0, 0.8);

0.9656854250 **(3)**

PlantGseq2(0.6, 0.3, 2.0, 0.8);

1.099677335 **(4)**