```
> #Nikita John, Attendance quiz 2
> aStupid := proc(n)
  if n = 0 then
  0 :
  elif n = 1 then
  1 :
  elif n = 2 then
  4 :
  else
  3·a(n − 1) − 3·a(n − 2) + a(n − 3) :
  fi:
  end:
  #I tried running it without option remember and it took way too long but here is the code I wrote
> #a1 = 0, a2 = 8, a3 = 4
  dsolve( { diff(y(t), t) = 0·t^8 / y(t)^4, y(1) = 8 }, y(t) )
```

$$y(t) = 8 \tag{1}$$

```
> #a1 = 6, a2 = 21, a3 = 50
  dsolve( {6· D(D(y))(t) + 21·D(y)(t) + 50·y(t) = 0, y(0) = 1, D(y)(0) = 0}, y(t) )
```

$$y(t) = \frac{7\sqrt{759}\, e^{-\frac{7t}{4}} \sin\left(\frac{\sqrt{759}\, t}{12}\right)}{253} + e^{-\frac{7t}{4}} \cos\left(\frac{\sqrt{759}\, t}{12}\right) \tag{2}$$

```
> with(LinearAlgebra);
```

[*&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, ProjectionMatrix, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix,* **(3)**

SylvesterSolve, *ToeplitzMatrix*, *Trace*, *Transpose*, *TridiagonalForm*, *UnitVector*, *VandermondeMatrix*, *VectorAdd*, *VectorAngle*, *VectorMatrixMultiply*, *VectorNorm*, *VectorScalarMultiply*, *ZeroMatrix*, *ZeroVector*, *Zip*]

```
> #a1 = 52, a2 = 50, a3 = 19
  A := Matrix( [ [52, 50, 19], [50, 19, 52], [19, 50, 52] ]);
```

$$A := \begin{bmatrix} 52 & 50 & 19 \\ 50 & 19 & 52 \\ 19 & 50 & 52 \end{bmatrix} \tag{4}$$

```
> Digits := 10;
```

$$Digits := 10 \tag{5}$$

```
> evalf( Eigenvalues(A) );
```

$$\begin{bmatrix} 121. \\ 33. \\ -31. \end{bmatrix} \tag{6}$$

```
> evalf( Eigenvectors(A) );
```

$$\begin{bmatrix} -31. \\ 121. \\ 33. \end{bmatrix}, \begin{bmatrix} 1. & 1. & -1.036153290 \\ -2.040000000 & 1. & 0.01373825018 \\ 1. & 1. & 1. \end{bmatrix} \tag{7}$$

```
>
```