```
> #Hrudai Battini Lec 10 Attendance
  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, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip] **(1)**

```
> #1 Who made fractals popular? Benoit Mandelbrot
> #2 Find Eigenvectors for the matrix from class. Find the first 20
  iterations.
  A := Matrix([[-3/2, 3/2],[-3, 11/4]]);
  Lambda := Eigenvalues(A);
  Evectors := Eigenvectors(A);
  d := evalf(A^20);
```

$$A := \begin{bmatrix} -\frac{3}{2} & \frac{3}{2} \\ -3 & \frac{11}{4} \end{bmatrix}$$

$$\Lambda := \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \end{bmatrix}$$

$$Evectors := \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \end{bmatrix}, \begin{bmatrix} \frac{2}{3} & \frac{3}{4} \\ 1 & 1 \end{bmatrix}$$

$$d := \begin{bmatrix} -0.02536111244 & 0.01902154959 \\ -0.03804309918 & 0.02853327806 \end{bmatrix} \tag{2}$$

>