```
"""
Registration: xxxx;
Description: Matrix Diagonalization, Inverse, Eigen vector/value.
             3 Identical Mass-Spring System ( 2  -1   0)(x1)         (x1)
                                            (-1   2  -1)(x2) = mw^2/k(x2)
                                            ( 0  -1   2)(x3)         (x3)
Author: AKB
"""

import numpy as np

# Four identical spring constant, three identical mass
k, m = 1.0, 1.0

a, b = input("number of rows and columns of A: ").split()
a, b = int(a), int(b)
A = np.array([[float(input("A"+str(i)+str(j)+" : ")) for j in range(b)] for i in
range(a)])

# Matrix Inverse
Ainv = np.linalg.inv(A)
print ('1/A = \n', Ainv)

# Identity Matrix
I = np.dot(A, Ainv)
print ('Orthogonality check: A*1/A = \n', I)

# Eigenvalues and Eigenvectors
eigen_val, eigen_vec = np.linalg.eig(A)
#eigen_val = np.linalg.eigvals(A)

# Print Results
print ('Eigen value of A = \n', eigen_val)
print ('Eigen vector of A = \n', eigen_vec)
print ('Eigen Frequencies are = \n', np.sqrt(k/m*eigen_val))

#for i in range(a):
#    print ('Eigen vectors of A = \n', eigen_vec[:,i])

"""
 Results
 number of rows and columns of A: 3 3
 A00 : 2
 A01 : -1
 A02 : 0
 A10 : -1
 A11 : 2
 A12 : -1
 A20 : 0
 A21 : -1
 A22 : 2
 1/A = [[ 0.75  0.5   0.25]
        [ 0.5   1.    0.5 ]
        [ 0.25  0.5   0.75]]
 A*1/A = [[  1.00000000e+00   0.00000000e+00   0.00000000e+00]
          [  5.55111512e-17   1.00000000e+00  -1.11022302e-16]
          [ -1.11022302e-16  -2.22044605e-16   1.00000000e+00]]
 Eigen value of A =      [ 3.41421356  2.          0.58578644]
 Eigen Frequencies are = [ 1.84775907  1.41421356  0.76536686]
 Eigen vector of A = [[ -5.00000000e-01  -7.07106781e-01   5.00000000e-01]
                     [  7.07106781e-01   4.05925293e-16   7.07106781e-01]
                     [ -5.00000000e-01   7.07106781e-01   5.00000000e-01]]
"""
```