```
%quantum logic gates
I = [1 0; 0 1]; X = [0 1; 1 0]; Y = [0 1j; -1j 0]; Z = [1 0; 0 -1];
H = [1 1; 1 -1]; Cx = [1 0 0 0; 0 1 0 0; 0 0 0 1; 0 0 1 0];
II = kron(I,I);XX = kron(X,X);YY = kron(Y,Y);ZZ = kron(Z,Z);

%will need 2 qubits for the 4x4 matrix
GivenMatrix = [1 0 0 0; 0 0 -1 0; 0 -1 0 0; 0 0 0 1];

%easy to see here given matrix expressed in basis{II,XX,YY,ZZ}
hamiltonian = 1/2*(II - XX - YY + ZZ)
```

```
hamiltonian = 4×4
     1     0     0     0
     0     0    -1     0
     0    -1     0     0
     0     0     0     1
```

```
%looking at lowest eigenvalue and corresponding eigenvector
[eigvec, eigval] = eig(hamiltonian)
```

```
eigvec = 4×4
         0         0    1.0000         0
   -0.7071   -0.7071         0         0
   -0.7071    0.7071         0         0
         0         0         0    1.0000
eigval = 4×4
    -1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
%need to make [0 1 1 0] i.e. ( ket{01} + ket{10} ) / sqrt(2)
%this is a bell state, maximally entangled
% https://en.wikipedia.org/wiki/Bell_state#Creating_Bell_states

Rx = @(x) (cos(x/2)*I - 1j*sin(x/2)*X);
IRx = kron(I,Rx(pi))
```

```
IRx = 4×4 complex
   0.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 1.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i
```

```
HI = kron(H,I);
Cx = [1 0 0 0; 0 1 0 0; 0 0 0 1; 0 0 1 0];

requiredEigenvector = real(Cx*HI*IRx*[1;0;0;0] *1j) %upto an overall phase
```

```
requiredEigenvector = 4×1
     0
     1
     1
     0
```

```
% a "bell state maker" will work as ansatz. Yay.
```