Using the Heisenberg Model Implementation

1. Make sure all libraires in listed in requirements.txt are satisfied
2. Store heisenbergModel.py in current directory then use, import heisenbergModel
3. Create heisenbergModel object and specify variables,

x = heisenbergModel(N,exchange,mcSteps,heisenberg, magneticField, plotBool, monteCarlo)

* N (int): Number of particles
* Exchange (int): determines magnetism of lowest energy configuration (>0:AFM, <0:FM)
* mcSteps (int): Number of Monte Carlo steps, initialized as zero
* heisenberg (Bool): Determines Heisenberg or Ising model, initialized as True (True:Heisenberg, False:Ising)
* magneticField(arr): Components of applied magnetic field initialized as None
* plotBool(Bool): Determines whether visual representation of final lattice is saved, initialized as False
* monteCarlo(Bool): Detemines whether plot of Monte Carlo Energies is calculated, initlaized as False. WARNING: Much higher computation time if True

1. We can now run simulations in all implemented dimensions by calling the corresponding method and inputting shape of the lattice if applicable: x.onedimension(), x.twoDimensions([i,j]), x.threeDimensions([i,j,k]) (Product of (i,j) or (i,j,k) must be equal to N)