```
using ITensors
using DelimitedFiles
OPEN_BOUND = 0
CLOSE_BOUND = 1
#dimension of the site of a boson is given by # of particle + 1
#
#|0>, |1> , |2> , |3>, |4>....|10>, |11>
#start off: 10 particles,

#|3>-> 2 particles

function
DMRG_Calculation(L::Int64,J::Float64,U::Float64,mu::Float64,V::Float64,boundCond,con
serveQn=true,d=30)
        sites = siteinds("Boson",L; conserve_qns=conserveQn, dim=d) #create L # of
sites of type bosons with dimension=d
        @time begin
        ampo = OpSum()
        for i in 1:(L-1)
            ampo += -J, "A", i, "Adag", i + 1
            ampo += -J, "Adag", i, "A", i + 1
            ampo += V, "N", i, "N", i + 1
        end
        for i in 1:L
            ampo += -mu, "N", i
            ampo += -U/2, "N", i
            ampo += U/2, "N", i, "N", i
        end
        if boundCond == CLOSE_BOUND
            ampo += -J, "A", L, "Adag", 1
            ampo += -J, "Adag", L, "A", 1
            ampo += V, "N", L, "N", 1
        end
        H = MPO(ampo,sites)
    end
    @time begin
        state = [3 for i in 1:L] #initialize our initial state.
        psi0 = randomMPS(sites, state, 10)
        @show flux(psi0)
        sweeps = Sweeps(5)
        setmaxdim!(sweeps, 10,50,100,200,400)  #m
        setcutoff!(sweeps, 1E-8)
        energy, psi = dmrg(H,psi0, sweeps)
    end
    return energy, psi
end

function Expect_N_Values(psi)
    values = expect(psi, "N")
    return values
```

```
end

function Expect_NN_Values(psi)
    values = correlation_matrix(psi,"N","N")
    return values
end

function Expect_AAdag_Values(psi)
    values = correlation_matrix(psi, "A", "Adag")
    return values
end

function Entanglement_Entropy(psi, b::Int64)
    orthogonalize!(psi, b)
    U,S,V = svd(psi[b], (linkind(psi, b-1), siteind(psi,b)))
    SvN = 0.0
    for n=1:dim(S, 1)
      p = S[n,n]^2
      SvN -= p * log(p)
    end
    return SvN
end

function Central_Charge(psi, L)
    entropy1 = Entanglement_Entropy(psi, Int(L/2)-1)
    entropy2 = Entanglement_Entropy(psi, Int(L/2))
    return 3*(entropy1-entropy2)/log(cos(pi/L))
end

L, J, U, mu, V = 20, 0.2, 1.0, 0.5, 0.0
energy, psi = DMRG_Calculation(L, J, U, mu, V, CLOSE_BOUND, true)
#N_values = Expect_N_Values(psi)
#display(sum(N_values))
#NN_values = Expect_NN_Values(psi)
#AAdag_values = Expect_AAdag_Values(psi)
#display(N_values)
#display(AAdag_values)

c = Central_Charge(psi, L)
display(c)
#writedlm( "AAdag_values_10_L=20.csv",  AAdag_values, ',')
```