```
# construct two point correlation matrix for HD
def construct_CM(L):
    Neel = np.zeros((int(L),int(L/2)))
    for i in range(0,int(L)):
        for j in range(0,int(L/2)):
            if i+1==2*(j+1)-1: Neel[i,j]=1
    CM = np.dot(Neel,Neel.transpose())          # CM in Ising basis
    return CM

# calculate LE using |det(1-C+C*exp(-iHt))|
def calc_detLE(v,U,CM,t):
    LE=np.zeros(len(t))
    for i in t:
        Ut = construct_U(v,U,i)
        k=t.tolist().index(i)
        LE[k]=np.abs(np.linalg.det(np.identity(args.L)-CM+np.dot(CM,Ut)))
    return LE


# Run the program
t=np.arange(args.tint,args.tmax+args.dt/2,args.dt)

# calculate part the single-particle Hamiltonian
SPH = construct_SPH(args.L,args.openbc)
vs,Us = np.linalg.eigh(SPH)
CM = construct_CM(args.L)

Store1=0
for samp in range(int(args.sample)):
    APDW = construct_APDW(args.L,args.W)
    SPHfW = SPH + APDW
    vsf,Usf = np.linalg.eigh(SPHfW)
    Store1 += calc_detLE(vsf,Usf,CM,t)

LE1=np.squeeze(Store1/args.sample)
RR1=-2*np.log(LE1)/args.L

for item in RR1:
    print(item)
```