```
#include<iostream>
#include <stdio.h>
#include <time.h>
#include<fstream>
#include<math.h>
#include<stdlib.h>
#include<vector>
using namespace std;

// based on P. Bracken, Int. J. Quant. Chem. 62 13 (1997).

void init(double & rho, int & n_prtcl, int & n_site, int & n_iter, double & Phi);
void init_kvc(int & n_prtcl, int & n_site, double & pi,double k_vec[]);
void calc_psi(int & n_prtcl, double & rho, double k_vec[], double *psi[]);
void calc_k_vec(int & n_prtcl, int & n_site, double k_vec[], double *psi[], double & Phi);
double calc_nrg(int & n_prtcl, int & n_site, double & rho, double k_vec[]);

int main()
{
  double rho,Phi;
  int n_prtcl,n_site,n_iter;

  init(rho,n_prtcl,n_site,n_iter,Phi);

  double pi = 4.*atan(1.);
  double *k_vec;
  double **psi;
  k_vec = new double[n_prtcl];
  psi = new double *[n_prtcl];
  for (int i_prtcl = 0; i_prtcl<n_prtcl;i_prtcl++)
    {
      psi[i_prtcl] = new double [n_prtcl];
    }

  init_kvc(n_prtcl,n_site,pi,k_vec);

  for (int i_iter=0;i_iter<n_iter;i_iter++)
    {
     calc_psi(n_prtcl,rho,k_vec,psi);
     calc_k_vec(n_prtcl,n_site,k_vec,psi,Phi);
    }
  cout << calc_nrg(n_prtcl,n_site,rho,k_vec);
     cout << "\n";

}

void init(double & rho, int & n_prtcl, int & n_site, int & n_iter, double & Phi)
{

  cout << "Rho (anisotropy):";
  cin >> rho;
  cout << "Phi (Peierls phase):";
  cin >> Phi;
  cout << "N_prtcl (particle number):";
  cin >> n_prtcl;
  cout << "N_site (number of sites):";
  cin >> n_site;
  cout << "N_iter (iterations):";
```

```
  cin >> n_iter;

}


void init_kvc(int & n_prtcl, int & n_site, double & pi, double k_vec[])
{

  double k0;

  k0 = pi*.5 - pi;

  for (int i_prtcl=0;i_prtcl<n_prtcl;i_prtcl++)
    {
      k_vec[i_prtcl] = k0 + pi/n_site*(2*i_prtcl+1);
    }

}


void calc_psi(int & n_prtcl, double & rho, double k_vec[], double *psi[])
{

  double cti,ctj;

  for (int i_prtcl=0;i_prtcl<n_prtcl;i_prtcl++)
    {
    for (int j_prtcl=0;j_prtcl<n_prtcl;j_prtcl++)
      {
        cti = cos(k_vec[i_prtcl]*0.5)/sin(k_vec[i_prtcl]*0.5);
        ctj = cos(k_vec[j_prtcl]*0.5)/sin(k_vec[j_prtcl]*0.5);
            psi[i_prtcl][j_prtcl] = 2.*atan(((1.+rho) - (1.-rho)*cti*ctj)/(cti-ctj)/rho);
      }
    }
}


void calc_k_vec(int & n_prtcl, int & n_site, double k_vec[], double *psi[], double & Phi)
{
  double pi = 4.*atan(1.);

  for (int i_prtcl=0;i_prtcl<n_prtcl;i_prtcl++)
    {
      k_vec[i_prtcl] = 2.*pi*(2*i_prtcl+1)/n_site + Phi;
    for (int j_prtcl=0;j_prtcl<i_prtcl;j_prtcl++)
      {
            k_vec[i_prtcl] = k_vec[i_prtcl] + psi[i_prtcl][j_prtcl]/n_site;
      }
    for (int j_prtcl=i_prtcl+1;j_prtcl<n_prtcl;j_prtcl++)
      {
            k_vec[i_prtcl] = k_vec[i_prtcl] + psi[i_prtcl][j_prtcl]/n_site;
      }
    }
}


double calc_nrg(int & n_prtcl, int & n_site, double & rho, double k_vec[])
{
```

```
  double nrg=0;

  for (int i_prtcl=0;i_prtcl<n_prtcl;i_prtcl++)
    {
      nrg = nrg + (cos(k_vec[i_prtcl])-rho)/n_site;
    }
  return nrg;
}
```