```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "nr.h"
#include "nrutil.h"
#include "nrutil2.c"
#define rand ran3(&ranint)

FILE *fout,*fin;
double ran3();
double Pi=3.14159;
double Kc=20.;
int ranint=100;
int nsteps=10000000;
int nfix;
int ncy=2;
double a=1.0;
double T=1.0;
double eb=50.;
double d2=.1;
double s3;
double stepsize=0.01;

/* Functions                                                    */
double distA(x,y) double x, y;
{
  int i;
  double l2,it;

  i=x/(3.*s3*a); it=x/(3.*s3*a); if (it-i >0.5) i++;
  l2=(x-3.*s3*a*i)*(x-3.*s3*a*i)+y*y;
  return l2;
}

double distB(x,y) double x, y;
{
  int i;
  double l2,it;

  i=(x-s3*a)/(3.*s3*a); it=x/(3.*s3*a); if (it-i  >0.5)  i++;
  l2=(x-3.*s3*a*i-s3*a)*(x-3.*s3*a*i-s3*a)+y*y;
  return l2;
}
```

```
double distC(x,y) double x, y;
{
 int i;
 double l2,it;

 i=(x-2.*s3*a)/(3.*s3*a);   it=x/(3.*s3*a); if (it-i >0.5) i++;
 l2=(x-3.*s3*a*i-2.*s3*a)*(x-3.*s3*a*i-2.*s3*a)+y*y;
 return l2;
}

double min(x,y) double x, y;
{return x<y ? x : y;}

main(ac, av) int ac; char **av;
{
 int n,i,j,*lct,A,B,C,R,nloop=1,k,dva;
 int son=0,tcount=0;
 double x, y, xA, xB, xC, yA, yB, yC, q, l2A, l2B, l2C, E=0., Et=0., Emin=0., w;
 double sx, sy, sxA, sxB, sxC, syA, syB, syC, sq, sl2A, sl2B, sl2C;
 double pA=1., pB=1., pC=1.,tt;
 double *rA, *rB, *rC, *cA, *cB, *cC;
 double xt, yt, xAt, xBt, xCt, yAt, yBt, yCt, qt;
 char fname[255];
 char fname1[255],fname2[255],fname3[255],fname4[255];

 if (ac>1) ranint = atof(av[1]);
 if (ac>2) Kc = atof(av[2]);
 if (ac>3) eb  = atof(av[3]);
 if (ac>4) d2  = atof(av[4]);
 if (ac>5) dva = atof(av[5]);

 s3=sqrt(3.); cA=dvector(1,1000); cB=dvector(1,1000); cC=dvector(1,1000);
 rA=dvector(1,1000); rB=dvector(1,1000); rC=dvector(1,1000);

 R=ranint;

 printf("ranint=%d, Kc=%g, eb=%g, T=%g, d2=%g\n",ranint,Kc,eb,T,d2);
 fin=fopen("lc.txt","r");
 for(i=1; i<=6*ncy; i++)
 {
  fscanf(fin,"%d %d %d",&A,&B,&C);
  rA[i] = Kc*A/(Kc+1.0); rB[i] = Kc*B/(Kc+1.0);  rC[i] = Kc*C/(Kc+1.0);
 }
```

```
/* set initial positions                                    */
x = (rand -0.5) + 5.; y = (rand -0.5) + 2.; q = rand * 2. * Pi;
xA = x + a * sin(q); yA = y + a * cos(q);
xB = x + a * sin(q-2*Pi/3); yB = y + a * cos(q-2*Pi/3);
xC = x + a * sin(q+2*Pi/3); yC = y + a * cos(q+2*Pi/3);
l2A = distA(xA, yA);  l2B = distB(xB, yB);  l2C = distC(xC, yC);

sx = x+3.*s3; sy = y; sq = rand * 2. * Pi;
sxA = sx + a * sin(sq); syA = sy + a * cos(sq);
sxB = sx + a * sin(sq-2*Pi/3); syB = sy + a * cos(sq-2*Pi/3);
sxC = sx + a * sin(sq+2*Pi/3); syC = sy + a * cos(sq+2*Pi/3);
sl2A = distA(sxA, syA);  sl2B = distB(sxB, syB);  sl2C = distC(sxC, syC);

if (rand<rA[1]) pA = 1.; else pA = 0.;
if (rand<rB[1]) pB = 1.; else pB = 0.;
if (rand<rC[1]) pC = 1.; else pC = 0.;

E = -1.* eb * (pA * exp(-1.*l2A/d2) + pB * exp(-1.*l2B/d2) + pC * exp(-1.*l2C/d2));

sprintf(fname,"twmKc%gV%gR%d.txt",Kc,eb,R);
fout=fopen(fname,"w+");

nfix=0.01/stepsize; nsteps=nsteps*nfix*nfix;

/* monte carlo steps                                        */
for (n=1; n<=nsteps; n++)
{
  if (rand<rA[j]) pA = 1.; else pA = 0.;
  if (rand<rB[j]) pB = 1.; else pB = 0.;
  if (rand<rC[j]) pC = 1.; else pC = 0.;

  l2A = distA(xA, yA);  l2B = distB(xB, yB);  l2C = distC(xC, yC);
  E = -1.* eb * (pA * exp(-1.*l2A/d2) + pB * exp(-1.*l2B/d2) + pC * exp(-1.*l2C/d2));

  random:
  xt = x + (rand -0.5) * stepsize;
  yt = y + (rand -0.5) * stepsize;
  qt = q + (rand -0.5)  *  stepsize;
  xAt = xt + a * sin(qt); yAt = yt + a * cos(qt);
  xBt = xt + a * sin(qt-2*Pi/3); yBt = yt + a * cos(qt-2*Pi/3);
  xCt = xt + a * sin(qt+2*Pi/3); yCt = yt + a * cos(qt+2*Pi/3);
  if (yt>2.5*a || yAt<=0 || yBt <=0 || yCt<=0) goto random;

  l2A = distA(xAt, yAt);  l2B = distB(xBt, yBt);  l2C=distC(xCt, yCt);
```

```
    Et = -1. * eb * (pA * exp(-1.*l2A/d2) + pB * exp(-1.*l2B/d2) + pC * exp(-1.*l2C/d2));

    w=min(1.,exp((E-Et)/T));
    if (rand <= w)
    {
      x = xt; y = yt; q = qt;
      xA = xAt; yA = yAt;
      xB = xBt; yB = yBt;
      xC = xCt; yC = yCt;
      E = Et;
    }
  }
}
```