```c
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include "nr.h"
#include "nrutil.h"

int main()
{
    FILE *filename, *filename2, *filename3, *filename4;
    char name_file[150], name_file2[150], name_file3[150], name_file4[150];

    long idum = -10;
    int s, p, t, temp, temp_a, ini, x_loc;
    double bc_limit=10.0, bin=0.1, n_x_temp;
    double dt=pow(bin,2)/20.0;
    int N_s = bc_limit/bin;
    int N_t=5000, N_p=0, ratio=500;


    int n[N_t+1][N_s+1];
    double x_coord[N_s+1];
    double t_coord[N_t+1];
    double u0[N_s+1];

    double q, delta_x;
    int energy[N_t+1], n_leave;


    // x-coord & u0
    for (s=0;s<=N_s;s++){
        x_coord[s] = bin * s ;
    }

    // t-coord
    t_coord[0]=0.0;
    for (t=0;t<=N_t;t++) t_coord[t] = dt * t;

    // initialization
    for (t=0;t<=N_t;t++){
        for (s=0;s<=N_s;s++){
            n[t][s] = 0;
        }
    }

    // initial condition
    for (s=1;s<=N_s;s++){
        n[0][s] = ratio;
        N_p += n[0][s];
    }
    printf("%d\n",N_p);
    double x[2][N_p+1];

    energy[0] = N_p;


    int tt=0;
    //Monte Carlo
    for (t=1;t<=N_t;t++){
```

```
int k_ini=0, k=0;

for (s=0;s<=N_s;s++){

    // n to x
    temp =0;
    k_ini = k;
    while (temp < n[t-1][s]){
        x[t-tt-1][k] = x_coord[s];
        //printf("%d\t%d\t%d\t%f\t%f\n",s, n[0]
            [s],temp,x_coord[s],x[0][k]);
        temp++;
        k++;
        //printf("%d\n",k);
    }
    //printf("k_ini: \t");
    //printf("%d\t%d\t%d\n",s ,temp ,k_ini);


    // evolve x
    for (p=k_ini;p<k;p++){

        if (s==0) q = gasdev(&idum) - ( (n[t-1][1] - n[t-1][0])/ratio/
            bin - 0.5);
        else q = gasdev(&idum);

        delta_x = q * sqrt( 2*dt );
        x_loc = round( ( x[t-tt-1][p]+delta_x ) / bin );

        if (x_loc >= 0 && x_loc<= N_s){
            x[t-tt][p] = x_coord[ x_loc ];

            // x to n
            n[t][x_loc]++;

            //printf("particle:\t");
            //printf("%d\t%d\t%f\t%d\t%f\t%f\t%d\n", t, p, delta_x,
                x_loc, x[t-tt-1][p], x[t-tt][p], n[t][x_loc]);
            //printf("%d\n",N_p);
        }
        else if (x_loc > N_s){
            x_loc = s;
            x[t-tt][p] = x_coord[ s ];

            n[t][x_loc]++;

        }
        else if (x_loc < 0 && s != 0 ) {
            x_loc = s;
            x[t-tt][p] = x_coord[ s ];

            n[t][x_loc]++;
        }
        else n_leave++;

    }
}
```

```
    tt=t;



    // number of particle
    N_p=0;
    for (s=0;s<=N_s;s++){
        N_p+=n[t][s];
        //printf("%d\n",N_p);
    }

    double x[2][N_p+1];

    for (s=0;s<=N_p;s++){
        x[0][s]=-9999.;
        x[1][s]=-9999.;
    }

    energy[t] = N_p + n_leave;



}


// output data
strcpy (name_file, "/Users/natii/nati/UIUC/2014Fall/CPA/ps9/q2c/data/
    n1.dat");
filename = fopen (name_file, "w");

for (s=0;s<=N_s;s++) {
    for (t=0;t<=N_t;t++){
        fprintf(filename, "%d\t", n[t][s]);
    }
    fprintf(filename, "\n");
}
fclose (filename);


strcpy (name_file2, "/Users/natii/nati/UIUC/2014Fall/CPA/ps9/q2c/data/
    t1.dat");
filename2 = fopen (name_file2, "w");

for (t=0;t<=N_t;t++) {
    fprintf(filename2, "%d\t%f\n",t ,t_coord[t]);
    //printf("%f\t",t[i]);
}
fclose (filename2);

strcpy (name_file3, "/Users/natii/nati/UIUC/2014Fall/CPA/ps9/q2c/data/
    x1.dat");
filename3 = fopen (name_file3, "w");

for (s=0;s<=N_s;s++) {
    fprintf(filename3, "%d\t%f\n",s ,x_coord[s]);
}
fclose (filename3);
```

```
    strcpy (name_file4, "/Users/natii/nati/UIUC/2014Fall/CPA/ps9/q2c/data/
        energy.dat");
    filename4 = fopen (name_file4, "w");

    for (t=0;t<=N_t;t++) {
        fprintf(filename4, "%d\t%d\n",t ,energy[t]);
    }
    fclose (filename4);



    return 0;

}
```