```
//Programme Labo: systÃ¨me auto-gravitant en deux dimensions
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_sort_double.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_sort_vector_double.h>
//temps associÃ© Ã  notre pas
#define dt 0.0001
#define pi 3.14

int main()
{ //DÃ©clarations
  FILE *pf1,*pf2,*pf3,*pf5;
  int nat;
  double tempx,tempy,r2,dx,dy,fi,ec,ep,temperature,pot,R;
  double e=0.001;
  double G=1.0;
  double m;
  char name[100];
  int i,j,maxpas,npas,ifixtemp,nouta,noutb,l;
  double *rx,*ry,*vx,*vy,*rax,*ray,*fx,*fy,*r;
  //Constantes
  double c0=dt,c1=dt*dt/2.0;
  double *rcx,*rcy,*f;
  double potj;
  //Lecture du fichier de ContrÃ´le
  pf1=fopen("CTRL.dat","r");
  fscanf(pf1,"%d  %d    %d  %lf\n",&maxpas,&nouta,&noutb,&m);
  fscanf(pf1,"%d  %lf   %d\n",&ifixtemp,&temperature,&nat);
  fclose(pf1);
  printf("%d      %d    %d  %lf\n",maxpas,nouta,noutb,m);
  printf("%d      %lf   %d    \n",ifixtemp,temperature,nat);


  gsl_permutation * perm = gsl_permutation_alloc(nat);
  gsl_permutation * rank = gsl_permutation_alloc(nat);
  gsl_vector * r_gsl = gsl_vector_alloc(nat);
  rx=malloc(nat*sizeof(double));
  ry=malloc(nat*sizeof(double));
  rcx=malloc(nat*sizeof(double));
  rcy=malloc(nat*sizeof(double));
  vx=malloc(nat*sizeof(double));
  vy=malloc(nat*sizeof(double));
  rax=malloc(nat*sizeof(double));
  ray=malloc(nat*sizeof(double));
  fx=malloc(nat*sizeof(double));
  fy=malloc(nat*sizeof(double));
  r=malloc(nat*sizeof(double));
  f=malloc(nat*sizeof(double));
  double * large = malloc (nat * sizeof(double));
  //Lecture configuration initiale
  pf1=fopen("CONF.IN","r");
  pf2=fopen("VISU2.IN","w");
  for (i=0;i<nat;i++)
    {
      fscanf(pf1,"%lf   %lf  %lf  %lf\n",&rx[i],&ry[i],&vx[i],&vy
[i]);
      //printf("%lf     %lf  %lf  %lf\n",rx[i],ry[i],vx[i],vy[i]);
      //vx[i]=vx[i]*sqrt(nat);
      //vy[i]=vy[i]*sqrt(nat);
    }
  fclose(pf1);
  fclose(pf2);
  pf5= fopen("films","w" );
  fprintf(pf5,"set size square\n");
  fprintf(pf5,"pau= 0.0\n");
  //  fprintf(pf1, "set xr[0:sqrt(3.0)]\n");
  fprintf(pf5, "set xr[-1:1]\n");
  //  fprintf(pf1,"set yr[0:sqrt(3.0)/2.0]\n");
  fprintf(pf5,"set yr[-1:1]\n");
  fprintf(pf5,"set zr[0:1]\n");
  //fprintf(pf1,"set data styl point\n");
  fclose(pf5);

  //BOUCLE PRINCIPALE
  pf2=fopen("TRAJ2","w");

  for (i=0;i<nat;i++)
    { tempx=rx[i];
      tempy=ry[i];
      rx[i]=rx[i]+vx[i]*c0+fx[i]*c1;
      ry[i]=ry[i]+vy[i]*c0+fy[i]*c1;
      rax[i]=tempx;
      ray[i]=tempy;
      fx[i]=0.0;
      fy[i]=0.0;
      r[i] = rx[i] * rx[i] + ry[i] * ry[i];
    }



  for(npas=1;npas<=maxpas;npas++)
    {




      //printf("%d      %d    \n",npas,maxpas);
      //Calcul des nouvelles forces
      ep=0.0;
      for (i=0;i<nat;i++)
      {
        //printf("%d          \n",i);
        r[i] = rx[i] * rx[i] + ry[i] * ry[i];
        gsl_vector_set(r_gsl, i, r[i]);
      }

      gsl_sort_vector_index(perm, r_gsl);
      gsl_permutation_inverse(rank, perm);

      for (i=0;i<nat;i++)
      {

        double ri = gsl_vector_get(r_gsl, i);
        //printf("element = %d, value = %g, rank = %d \n", 
i,ri,rank->data[i]);

        // x,y coordonnÃ©e cartesienne-vecteur unitaire
        double x=1,y=1;
        //r chapeau x

        rcx[i]=rx[i]/sqrt(rx[i]*rx[i]+ry[i]*ry[i]);
        //r chapeau y
        rcy[i]=ry[i]/sqrt(rx[i]*rx[i]+ry[i]*ry[i]);


        fi=(-2.0*G*m*m*((double)(rank->data[i]) -1.0 )/sqrt(r[i]));

        fx[i]=fi*rcx[i];
        fy[i]=fi*rcy[i];

        //printf("%lf  %lf   %lf\n", fi, rx[i], ry[i]);

        //f[i]=fx[i]+fy[i];


        //nat-(rank->data[i]);
        //gsl_sort_largest (large,nat-(rank->data[i]), r, 1, nat);

        //    for (j = 0; j < nat-(rank->data[i]) ; j++)
        //{
        //   printf ("%d: %.18f\n", j, large[j]);
        //potj=-log(large[j])+potj;
        //}

        pot=2.0*G*m*m*(potj-((rank->data[i]-1)*log(r[i])));
        ep=ep+pot;

        //  printf("i = %d ri = %lf fi = %lf  fx = %lf  fy = %lf rcx
= %le rcy = %le rang = %le \n",i,r[i],fi,fx[i],fy[i],rcx[i],rcy[i],
(double) rank->data[i]);

      }
      //printf("%lf           %lf   \n",ep,dt*npas);

      ec=0.0;
      //IntÃ©gration des positions et vitesses
      for (i=0;i<nat;i++)
      { tempx=rx[i];
        tempy=ry[i];
        rx[i]=2.0*rx[i]+2.0*c1*fx[i]/m-rax[i];
        ry[i]=2.0*ry[i]+2.0*c1*fy[i]/m-ray[i];
        vx[i]=(rx[i]-rax[i])/(2*c0);
        vy[i]=(ry[i]-ray[i])/(2*c0);
        rax[i]=tempx;
        ray[i]=tempy;
        fx[i]=0.0;
        fy[i]=0.0;
        ec=ec+vx[i]*vx[i]+vy[i]*vy[i];
      }

      ec=ec*0.5;

      //Ecriture dans ENE


      if (npas%nouta==0)
      {
        printf("%d %d %d\n",npas,nouta,npas%nouta);
        if(npas == nouta)
          {
            pf1=fopen("ENE2","w");
            pf3=fopen("RAYON2","w");
          }
        else
          {
            pf1=fopen("ENE2","a");
            pf3=fopen("RAYON2","a");
          }

        fprintf(pf1,"%lf      %lf   %lf   \n",npas*dt,ec,ep);
        R=0.0;

        for (i=0;i<nat;i++)
```

```
          {
            R+=rx[i]*rx[i]+ry[i]*ry[i];
          }
        fprintf(pf3,"%lf      %lf \n", dt*npas,R);

        fclose(pf1);
        fclose(pf3);

      }
      //printf("%d      \n",npas);


      //Ecriture dans TRAJ
      if (npas%noutb==0)
      { for (i=0;i<nat;i++)
          fprintf(pf2,"%lf    %lf   \n",rx[i],ry[i]);
      }

      // R=0.0;

      //for (i=0;i<nat-1;i++)
      //{
      //  R+=rx[i]*rx[i]+ry[i]*ry[i];
      //    }
      //fprintf(pf3,"%lf      %lf \n", dt*npas,R);

      pf5=fopen(name,"w");
      sprintf(name, "cos_%03d", l );
      for (i=0;i<nat;i++)
      {
        fprintf(pf5,"%lf      %lf %lf %lf \n",rx[i],ry[i],vx[i],vy
[i]);
      }

      fclose(pf5);

      pf5= fopen("films","a");
      //  fprintf(pf5,"plot \'%s\' using 1:2,\'%s\' using 1:2 w p 3,
sqrt(3.0)*x w l 3, sqrt(3.0)*(x-1) w l 3\n",name,name2);
      fprintf(pf5,"plot \'%s\' using 1:2 w p\n",name);
      fprintf(pf5,"pause pau\n");
      fclose(pf5);



      l +=1;
    } // Fin de la boucle principale

  fclose(pf2);
  //Ecriture configuration initiale

  pf1=fopen("CONF2.OUT","w");
  pf2=fopen("VISU2.OUT","w");
  for (i=0;i<nat;i++)
    {
      fprintf(pf1,"%lf  %lf   \n",rx[i],ry[i]);
      fprintf(pf1,"%lf  %lf \n",vx[i],vy[i]);
      fprintf(pf2,"%lf  %lf   \n",rx[i],ry[i]);
    }
  fclose(pf1);
  fclose(pf2);

  gsl_permutation_free(perm);
  gsl_permutation_free
(rank);
```

```
  return 0;
}
```