```
  //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,*pf4,*pf5;
    int nat;
    double tempx,tempy,r2,dx,dy,fij,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;
    //Lecture du fichier de ContrÃ´le
    pf1=fopen("CTRL.dat","r");
    fscanf(pf1,"%d      %d    %d \n",&maxpas,&nouta,&noutb,&m);
    fscanf(pf1,"%d      %lf   %d\n",&ifixtemp,&temperature,&nat);
    fclose(pf1);
    printf("%d    %d    %d \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));
    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));


    //Lecture configuration initiale
    pf1=fopen("CONF.IN","r");
    pf2=fopen("VISU.IN","w");
    pf3=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);

      //fscanf(pf1,"%lf %lf   \n",&vx[i],&vy[i]);
      //fprintf(pf2,"%lf      %lf   \n",rx[i],ry[i]);

      }
    fclose(pf1);
    fclose(pf2);
```

```
    fclose(pf3);


    pf5= fopen("film","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

    pf1=fopen("ENE","w");
    pf2=fopen("TRAJ","w");
    pf4=fopen("RAYON","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];
      }

    l=0;

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


          for (j=i+1;j<nat;j++)
            {     dx=rx[i]-rx[j];
            dy=ry[i]-ry[j];
            r2=dx*dx+dy*dy+e*e;
            fij=-2.*G*m*m/r2;
            fx[i]=fx[i]+dx*fij;
            fy[i]=fy[i]+dy*fij;
            fx[j]=fx[j]-fij*dx;
            fy[j]=fy[j]-fij*dy;

            pot=2.0*G*m*log(sqrt(r2));
            ep=ep+pot;
            }
        }
```

```
      printf("%lf       %lf   \n",ep,dt*npas);
      ec=0.0;
      //IntÃ©gration des positions
      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];
        }
      //IntÃ©grations des vitesses

      ec=ec*0.5;

      //Ecriture dans ENE

      if (npas%nouta==0)
        {
          fprintf(pf1,"%lf    %lf   %lf   \n",npas*dt,ec,ep);

        }

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


      //fprintf(pf3,"%le  %le   %le   %le  %le %le  %le %le\n",rx
[0],ry[0],rx[1],ry[1],vx[0],vx[1],vx[1],vy[1]);
      R=0.0;

      for (i=0;i<nat-1;i++)
      {
        R+=rx[i]*rx[i]+ry[i]*ry[i];
      }
fprintf(pf4,"%lf  %lf \n", dt*npas,R);



 pf5=fopen(name,"w");
 sprintf(name, "coo_%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("film","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(pf1);
    fclose(pf2);
    fclose(pf4);

    //Ecriture configuration initiale

    pf1=fopen("CONF.OUT","w");
    pf2=fopen("VISU.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;
  }
```