

```

/* GRAFIC20.C Palline che rimbalzano. C'è l'attrito che le fa rallentare
/* Simulazione di urti elastici di [MAX 50] sfere di diverse dimensioni sulle pareti dello schermo e tra di loro
/* Il numero delle sfere, la posizione, la velocità iniziale e il raggio viene fissato a caso tramite la funzione Random()

#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

int mode,driver;
long int cont,i,j,k,colore[50],raggio[50],imax,ascissa,ordinata;
float x[50],y[50],kx[50],ky[50],raggiof[50];

/*-----Dichiarazione variabili per il calcolo del rimbalzo di due biglie tra di loro-----*/
float m1,m2,vcx,vcy,xq,v01p,v02p,v1xp,v1yp,v2xp,v2yp,xip,yip,xkp,ykp;
float cosv01p,sinv01p,sinc1c2,cosc1c2,distanzac1c2,singamma,cosgamma;

main()
{
    driver=DETECT;
    mode=0;
    initgraph(&driver,&mode,"");
    randomize();
    imax=random(30)+20;
    for (i=0;i<imax;i++)
    {
        kx[i]=((random(501)+400)/1000.001)*5.001;
        ky[i]=((random(501)+400)/1000.001)*5.001;
        raggiof[i]=random(2000)/100.001+5;
        raggio[i]=raggiof[i];
        x[i]=random(getmaxx()-2*raggio[i])+raggio[i];
        y[i]=random(getmaxy()-2*raggio[i])+raggio[i];
        colore[i]=random(getmaxcolor()-1)+1;
    }
    cont=0;
    do
    {
        cont++;
        for (i=0;i<imax;i++)
        {
            setcolor(0);
            ascissa=x[i];
            ordinata=y[i];
            circle(ascissa,ordinata,raggio[i]);
            kx[i]=kx[i]*0.998;
            ky[i]=ky[i]*0.998;
            x[i]=x[i]+kx[i];
            y[i]=y[i]+ky[i];
            if ((x[i]>getmaxx()-raggio[i])||(x[i]<raggio[i])){kx[i]=-kx[i];}
            if ((y[i]>getmaxy()-raggio[i])||(y[i]<raggio[i])){ky[i]=-ky[i];}

/*-----Calcolo del rimbalzo della sfera [i] con le altre non ancora prese in considerazione [i+1..imax]-----*/
            for (k=i+1;k<imax;k++)
            {
                xip=x[i]+kx[i];
                yip=y[i]+ky[i];
                xkp=x[k]+kx[k];
                ykp=y[k]+ky[k];
                if(((x[i]-x[k])*(x[i]-x[k])+(y[i]-y[k])*(y[i]-y[k]))>(raggiof[i]+raggiof[k])*(raggiof[i]+raggiof[k]))&&((xip-
                xkp)*(xip-xkp)+(yip-ykp)*(yip-ykp)<(raggiof[i]+raggiof[k])*(raggiof[i]+raggiof[k])))
                /* Si confrontano il quadrato della distanza dei due centri e quello della somma dei raggi nella posizione attuale e successiva */
                {
                    distanzac1c2=sqrt((x[i]-x[k])*(x[i]-x[k])+(y[i]-y[k])*(y[i]-y[k]));
                    m1=raggio[i]*raggio[i]*raggio[i];
                    m2=raggio[k]*raggio[k]*raggio[k];
                    vcx=(m1*kx[i]+m2*kx[k])/(m1+m2);

```

