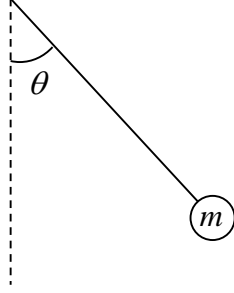


1) Aşağıdaki programın çıktısını yandaki tabloya yazınız.

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
int ff(int a,int b,int c);
int main()
{
int i=4,j,k,dd;
for(j=1;j<=2;j++){
for(k=2;k<=4;k++){
dd=ff(i,j,k);
if(i<(j+3) || k<=3){
printf("%d %d %d %d\n",i,j,k,dd);
i=i+dd;
}}}
return 0;
}
int ff(int a,int b,int c){
int d;
d=b*c-2*a;
return d;
}
```

i	j	k	dd
4	1	2	-6
-2	1	3	7
5	2	2	-6
-1	2	3	8

2) Şekilde verilen basit sarkaç için zamana bağlı olarak θ açısı değerini ve w açısal hızını Euler yöntemi ile bulup ekrana yazdıran programı yazınız. $l=0.5$ m , $g=9.81$ m/s² $t=0$ da $\theta =0.7$ rad ve $w =0$ rad/s dir.



$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin\theta = 0$$

$$w = \frac{d\theta}{dt} \quad \alpha = \frac{dw}{dt}$$

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
int main()
{
    double th[1000],w[1000],t[1000];
    double l,g,dt;
    int i;
    l=0.5;
    g=9.81;
    dt=0.01;
    th[0]=0.7;
    w[0]=0;
    for(i=0;i<=200;i++){
        t[i]=i*dt;
        th[i+1]=th[i]+w[i]*dt;
        w[i+1]=w[i]-g/l*sin(th[i+1])*dt;

        printf("%f\t%f\t%f\n",t[i],th[i],w[i]);
    }

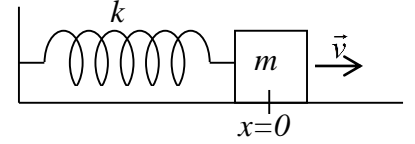
    return 0;
}
```


3) Şekildeki gibi kütle yay sistemine, hızına bağlı $0.2v$ büyüklüğünde sürtünme kuvveti etkimektedir. Kütle $t=0$ anında yayın denge konumunda ($x=0$) ve hızı sağa doğru $v = 10m/s$ değerindedir. Kütle yay sistemi için aşağıda verilen Verlet algoritmasını kullanarak, tüm zamanlarda konumu ve hızı bulup ekrana yazdıran bir program yazınız. ($m=0.5$ kg, $k=145$ N/m ve $\Delta t = 0.01$ dir.)

(Not; Kuvvetin bir sonraki değerini bulmak için $x(t + \Delta t)$, konumun bir sonraki değerini ve $v(t)$, hızın bir önceki değerini kullanabilirsiniz)

$$\text{Verlet algoritması: } x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}\Delta t^2 \frac{F(t)}{m}$$

$$v(t + \Delta t) = v(t) + \frac{F(t) + F(t + \Delta t)}{2m} \Delta t$$



#include <stdio.h>	0.000000	0.000000	10.000000
#include <stdlib.h>	0.010000	0.099800	9.815290
int main()	0.020000	0.196309	9.346670
{	0.030000	0.286743	8.608858
	0.040000	0.368501	7.624318
double x[1000],v[1000],t[1000],f[1000];	0.050000	0.439249	6.422583
double m,k,dt;	0.060000	0.496977	5.039365
int i;	0.070000	0.540064	3.515498
m=0.5;	0.080000	0.567318	1.895733
k=145;	0.090000	0.578011	0.227424
dt=0.01;	0.100000	0.571899	-1.440855
x[0]=0.0;	0.110000	0.549227	-3.060725
v[0]=10;	0.120000	0.510717	-4.585402
for(i=1;i<=100;i++){	0.130000	0.457550	-5.971047
f[i]=-k*x[i]-0.2*v[i];	0.140000	0.391324	-7.178030
x[i+1]=x[i]+v[i]*dt+0.5*dt*dt*f[i]/m;	0.150000	0.314013	-8.172056
f[i+1]=-k*x[i+1]-0.2*v[i];	0.160000	0.227903	-8.925146
v[i+1]=v[i]+(f[i]+f[i+1])*dt/2.0/m;	0.170000	0.135525	-9.416416
printf("%f\t%f\t%f\t%f\n",t[i],x[i],v[i]);	0.180000	0.039584	-9.632660
}	0.190000	-0.057124	-9.568697
	0.200000	-0.151791	-9.227496
return 0;	0.210000	-0.241680	-8.620053
}	0.220000	-0.324204	-7.765040
	0.230000	-0.396998	-6.688237
	0.240000	-0.457990	-5.421750
	0.250000	-0.505459	-4.003062
	0.260000	-0.538080	-2.473919
	0.270000	-0.554968	-0.879104
	0.280000	-0.555694	0.734871
	0.290000	-0.540302	2.321127
	0.300000	-0.509303	3.833770
	0.310000	-0.463657	5.229228
	0.320000	-0.404747	6.467496
	0.330000	-0.334332	7.513290
	0.340000	-0.254502	8.337046
	0.350000	-0.167608	8.915756
	0.360000	-0.076198	9.233612
	0.370000	0.017058	9.282430

