

Исследование физических моделей.
Движение тела, брошенного под углом к
горизонту.

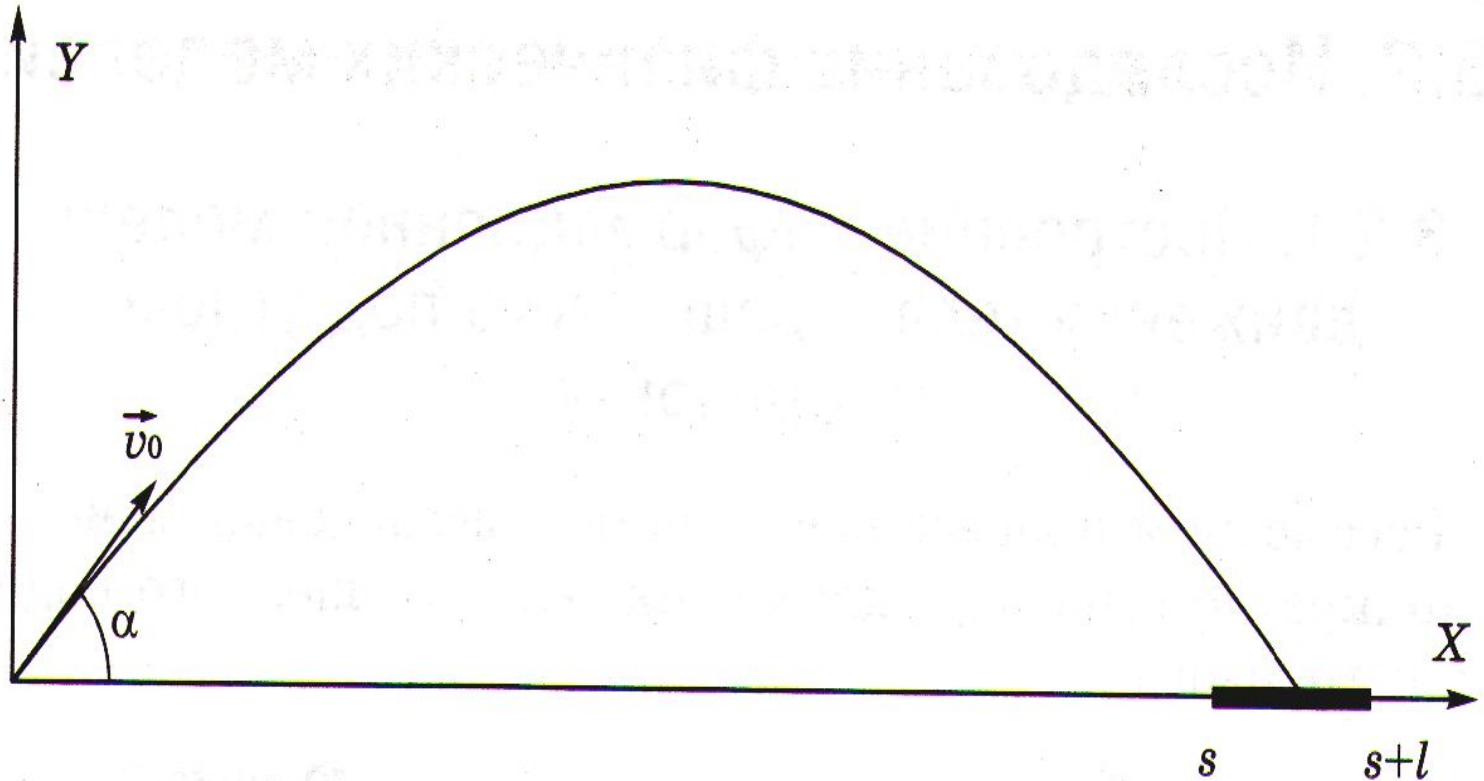
Бросание мяча в площадку

Попадет ли мяч, брошенный со скоростью 18 м/сек под углом 30° , в площадку длиной 1 м, находящуюся на расстоянии 25 метров?

Допущения:

- мяч считаем материальной точкой
- ускорение свободного падения $g=9,8 \text{ м/с}^2$
- движение по оси OY равноускоренное
- сопротивление воздуха не учитываем
- движение по оси OX равномерное

Графическая модель



Формальная (математическая) модель:

при заданных v_0 и a

$$x = v_0 \cdot \cos\alpha \cdot t;$$

$$y = v_0 \cdot \sin\alpha \cdot t - g \cdot t^2 / 2.$$

$y=0$ (площадка на земле)

$$v_0 \cdot \sin\alpha \cdot t - g \cdot t^2 / 2 = 0,$$

$$t \cdot (v_0 \cdot \sin\alpha - g \cdot t / 2) = 0.$$

$$v_0 \cdot \sin\alpha - g \cdot t / 2 = 0,$$

$$t = (2 \cdot v_0 \cdot \sin\alpha) / g.$$

Формальная (математическая) модель

$$x = (v_0 \cdot \cos\alpha \cdot 2 \cdot v_0 \cdot \sin\alpha) / g = (v_0^2 \cdot \sin 2\alpha) / g$$

Условие попадания мяча в площадку

$$S < x < S+L$$

где S - расстояние до площадки, L - длина площадки

Если $x < S$, то «недолет», если $x > S+L$, то «перелет».

Написать программу, с помощью которой представить рисунок площадки, траекторию движения мяча, задавая все необходимые начальные параметры, сделать текстовый вывод «Попадание», «Недолет», «Перелет».

Сделать вывод «Попадание», «Недолет», «Перелет».

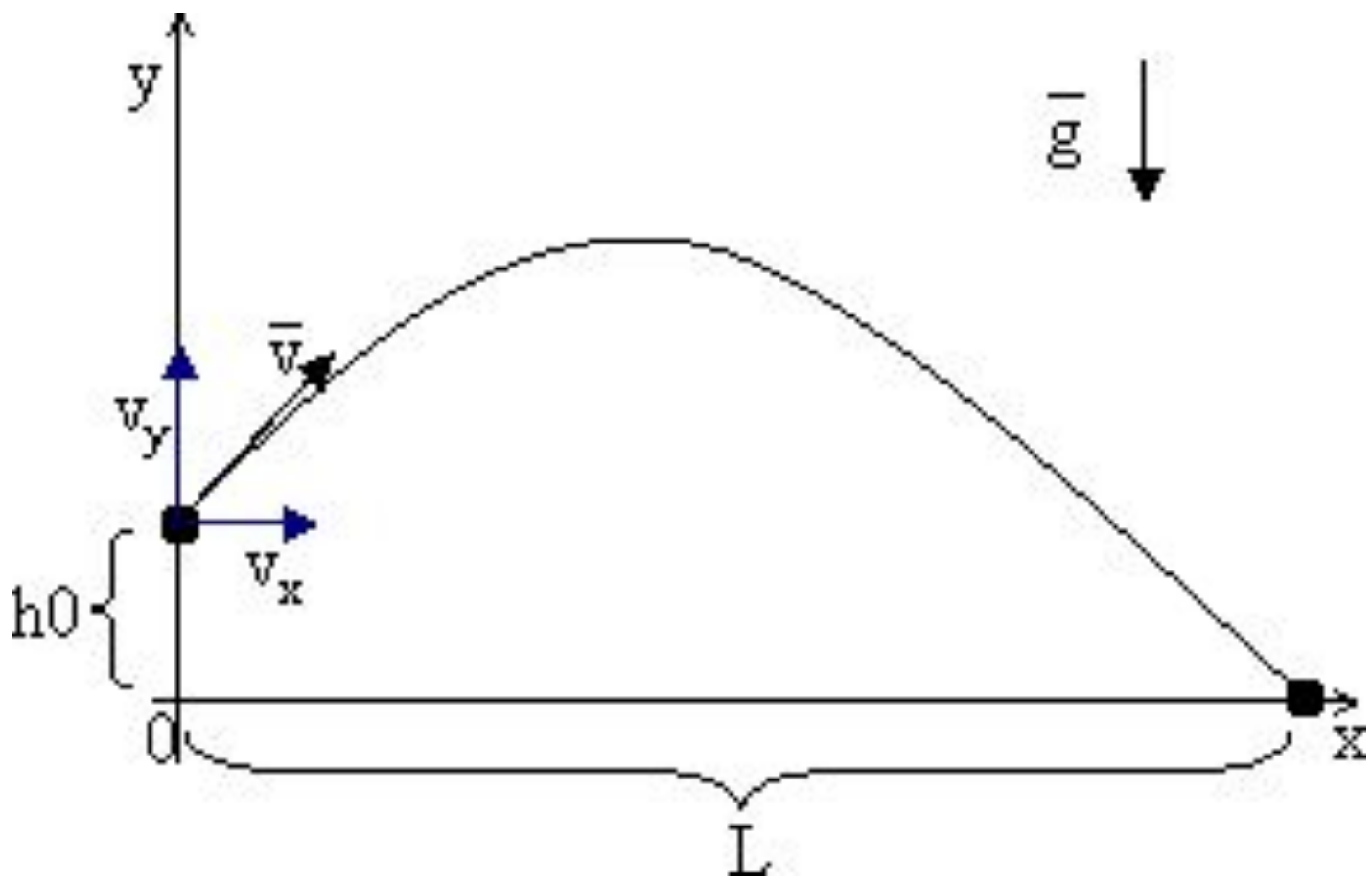
```
uses crt, graph;
var  driver,n:integer; str:string;  x,y,v,a,l,kx,ky,s:real;
begin
readln(s,L,v,a); driver:=detect; initgraph(driver,n,'e:\bp\bgi');
kx:=600/(s+l);  ky:=80;
line(20,440,620,440); line(20+trunc(kx*s),438,20+trunc(kx*(s+l)),438);
x:=0; y:=0.1;
while (y>0) and (not keypressed) and (kx*x<=620) do
begin
fillellipse(20+trunc(kx*x),438-trunc(ky*y),5,5);
delay(5000);
x:=x+0.1; y:=(x*v*v*sin(2*a*pi/180)-9.8*x*x)/(2*v*v*sqr(cos(a*pi/180)));
end
if (x>=s-0.1) and (x<=s+L+0.1) then outtextxy(320,240,'popal')
else if x>s+l+0.1 then outtextxy(320,240,'perelet')
else  outtextxy(320,240,'nedolet');
readln;
closegraph;
end.
```


Написать программу, которая позволит для каждого значения скорости бросания мяча получить с заданной точностью диапазон значений углов, обеспечивающих попадание мяча в площадку.

Получить диапазон углов для начальных параметров к задаче 1.

Задача 2.

Тело брошено с некоторой высоты с начальной скоростью, направленной под углом к горизонту. Определить угол, при котором дальность полета будет максимальной. Нарисовать траектории движения при разных параметрах.



$$x = v_0 t \cos \alpha \qquad y = y_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

$$0 = h_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

$$gt^2 - 2v_0 t \sin \alpha - 2h_0 = 0$$

$$D = (2v_0 \sin \alpha)^2 + 4g \cdot 2h_0$$

$$t_{1,2} = \frac{2v_0 \sin \alpha \pm \sqrt{(2v_0 \sin \alpha)^2 + 8gh}}{2g}$$

$$L = v_0 t \cos \alpha$$

```

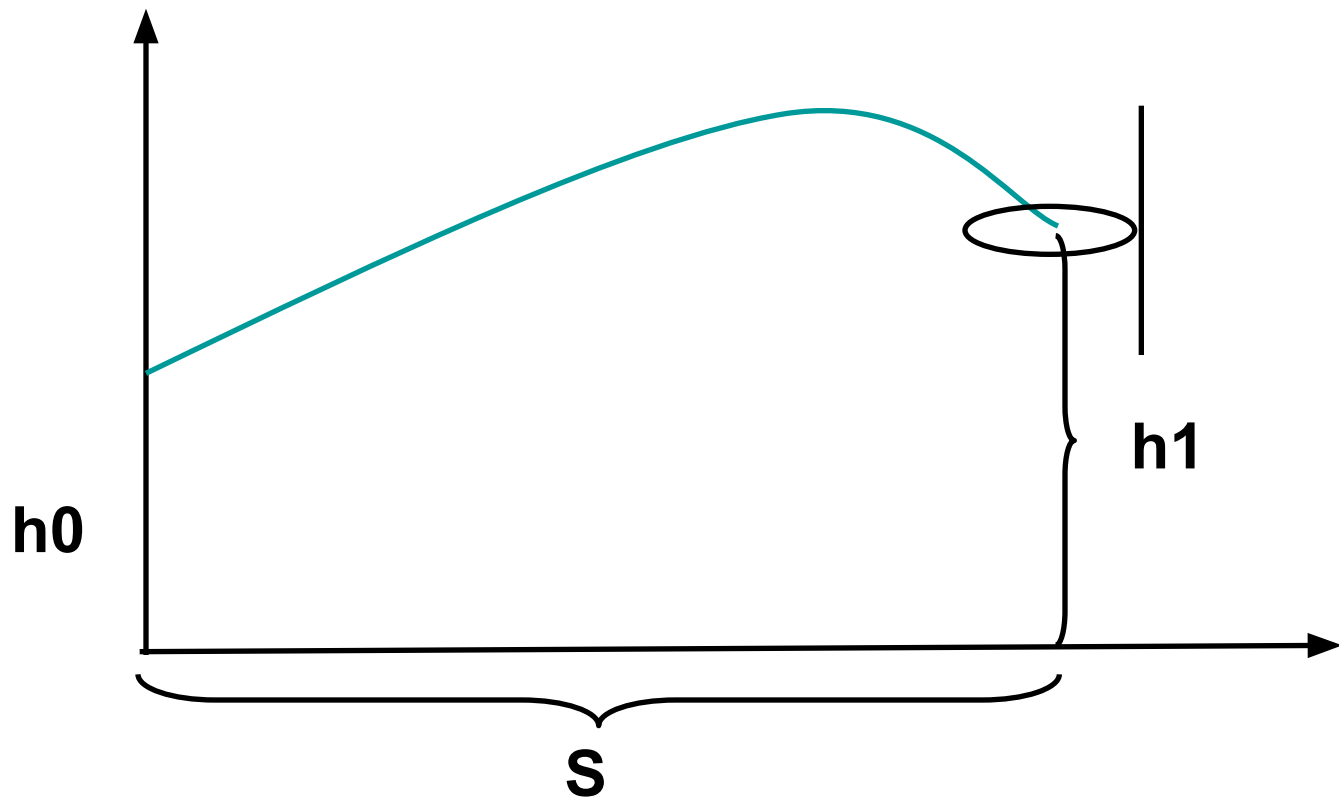
program polet;
uses crt, graph;
const g=9.8;
var  driver,n,a,nmax:integer; s:string;  h,x,y,v,kx,ky,l,max:real;
function y1(a:integer; x:real):real;
    begin
        y1:=x*sin(a*pi/180)/cos(a*pi/180)-(g/2*x*x)/(v*v*sqr(cos(a*pi/180)));
    end;
function t(v,h:real;a:integer):real;
begin
t:=(2*v*sin(a*pi/180)+sqr(sqr(2*v*sin(a*pi/180))+8*g*h))/2/g;
end;
begin
readln(v,h);
max:=0;
for a:=0 to 90 do
begin
l:=v*t(v,h,a)*cos(a*pi/180);
if l>max then begin max:=l; nmax:=a end;
end;
writeln(max:5:1, ' ', nmax);
readln;

```

```
driver:=detect;
initgraph(driver,n,'e:\bp\bgi');
kx:=20; setbkcolor(9);
ky:=20;
line(20,240,620,240);
y:=0; x:=0;
while (y>-h) and (not keypressed) do
begin
fillellipse(20+trunc(kx*x),240-trunc(ky*y),5,5);
delay(5000);
x:=x+0.1; y:=y1(nmax,x);
end;
str((x-0.1):4:1,s);
outtextxy(240,320,s);
readln;
closegraph;
end.
```

Задача 3

- Определить начальные параметры (скорость и угол), при которых следует бросить баскетбольный мяч в кольцо.
- Мяч должен попасть в кольцо «навесом», т.е. после прохода наивысшей точки.
- Начальная скорость до 15 м/с, кольцо на высоте 3м, расстояние до кольца от 0,5 до 7 м, точность попадания 20 см.



$$x = v_0 t \cos \alpha \qquad y = y_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

$$h_1 = h_0 + v_0 t \sin \alpha - \frac{gt^2}{2}$$

```

program polet;
uses crt, graph;
const g=9.8;
var  driver,n,a,m,p,nmax:integer; h1,t,t0,y0,h0,x,x1,y,y2,v,kx,ky,s:real;
function y1(a:integer; x:real):real;
    begin
        y1:=x*sin(a*pi/180)/cos(a*pi/180)-(g/2*x*x)/(v*v*sqr(cos(a*pi/180)));
    end;
begin
readln(v,h0,h1,s); y2:=0;
for a:=1 to 89 do begin
    t:=s/v/cos(a*pi/180); t0:= (s+0.1)/v/cos(a*pi/180);
    y:=h0+v*t*sin(a*pi/180)-g*sqr(t)/2;
    y0:=h0+v*t1*sin(a*pi/180)-g*sqr(t)/2;
    if (abs(y-h1)<=0.1)and(y-y0>0) then nmax:=a;
        end;
writeln(nmax); readln;

```

```
driver:=detect; initgraph(driver,n,'e:\bp\bgi');
kx:=20; ky:=20; line(20,240,620,240);
y:=0; x:=0; m:=0;
while (m=0) and (not keypressed) do
begin
fillellipse(20+trunc(kx*x),240-trunc(ky*y),5,5);
delay(5000); y2:=y;
x:=x+0.1; y:=y1(nmax,x);
if (y-y2<0)and (y<=h1-h0+0.1) then m:=1;
end;
readln;
closegraph;
end.
```