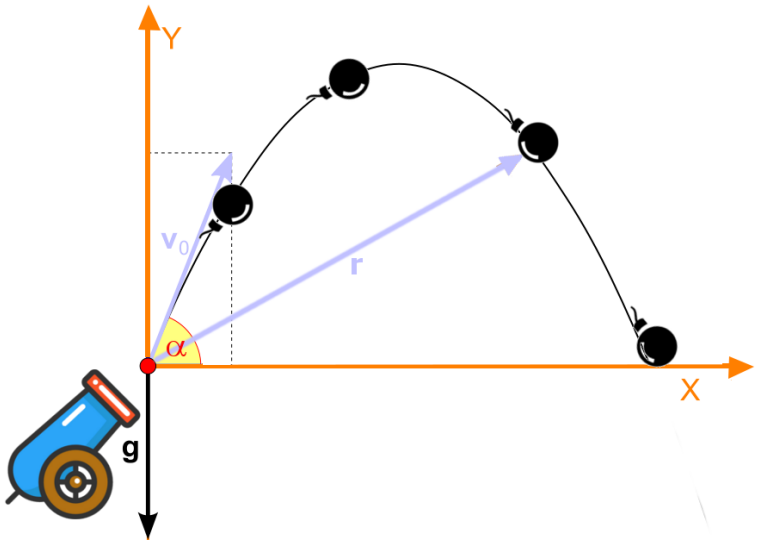
FELADAT

Egy ágyúból kilőtt golyó `v0` (m/s) kezdeti sebességgel, `alfa` (fok) szöggel indul.



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

$$y = v_0 \cdot t \cdot \sin\alpha - \frac{g}{2} \cdot t^2$$

Írj programot, amely bekéri ezeket az adatokat. Írja ki, hogy `t=0,0; 0,1; 0,2; …` s időpontokban (tized másodpercenként) az ágyúgolyó helyét; egészen addig, amíg be nem csapódik az a földbe (`y≤0`)!

Ehhez a `math.h` függvényei, a `sin()` és a `cos()` használhatóak. Vigyázz arra, hogy a szöget ezeknek a függvényeknek radiánban kell megadni! *Megjegyzés*: 1° = π/180 rad, g = 9,81 m/s^2

Végezd el az alábbi műveleteket!

a. Írasd ki a programmal, hogy mikor volt a legmagasabban az ágyúgolyó?
b. Ez mikor következett be?
c. Mikor csapódott be a földbe?
d. Határozd meg az ágyúgolyó legtávolabbi pontját!
e. Készíts grafikont az ágyúgolyó mozgásáról!

FELADAT MEGOLDÁSÁNAK LÉPÉSEI

I. *Megírtuk az alap programot:*

```
#include <iostream>
#include <math.h>

using namespace std;

int main()
{
  setlocale(LC_ALL, "");

  double v0, alfa, t, rad, x, y, r, g = 9.81;

  cout << "Ágyúgolyó mozgása\n\n";
  cout << "Kérem az adatokat! (ágyú golyó kezdeti sebessége,
          illetve kilövés szöge)\n\n";
  cout << "\tv0 [0-200 m/s] = ";
  cin >> v0;
  cout << "\talfa [0-90°] = ";
  cin >> alfa;
  cout << endl;
  rad = alfa*M_PI/180; szöget átváltjuk radiánba
  t = 0;
  x = 0;
  y = 0;
  cout << "\t t \t x \t y \n";
  while ( y >= 0 ) addig ismétlődik, amíg be nem csapódik
  {
    cout << "\t" << t << "\t" << x << "\t" << y << "\t" << endl;
    t = t + 0.1;
    x = v0*t*cos(rad);
    y = v0*t*sin(rad)-(g/2)*t*t;
  }
  cout << endl;
  cout << t << " másodpercben már biztosan becsapódott.";
  return 0;
}
```

szükséges adatokat változókban tároljuk

bekérjük az adatokat a felhasználótól

jelezzük, hogy milyen adatokat várunk

elvégezzük a szükséges számításokat

II. *Módosítások, kiegészítések:*

1. Adatbeolvasás ellenőrzés

Azaz csak a megadott tartományba eső adatokat fogadjon el a program!

```
do
{
  cout << "\tv0 [0-200 m/s] = ";
  cin >> v0;
  if ( v0 < 0 || v0 > 200 )
    cout << "\n\tNem jó adat, kérem újra!\n\n";
}
while( v0 < 0 || v0 > 200);

do
{
  cout << "\n\talfa [0-90°] = ";
  cin >> alfa;
  if ( alfa < 0 || alfa > 90 )
    cout << "\n\tNem jó adat, kérem újra!\n";
}
while( alfa < 0 || alfa > 90 );
```

2. Adatok tárolása

Adatokkal műveleteket szeretnénk végezni (pl. maxkiválasztás) => eltároljuk egy struktúra típusú tömbben.

```
struct helyzet
{
    double x,y,t;
};

helyzet tomb[1000];
int db = 0;

while ( y >= 0 )
{
  t = t + 0.1;
  x = v0*t*cos(rad);
  y = v0*t*sin(rad)-(g/2)*t*t;
  tomb[db].t = t;
  tomb[db].x = x;
  tomb[db].y = y;
  db ++;
}
db --;
```

számítások eredményét letároljuk a tömbben

3. Legmagasabb pont megkeresése

```
int lm = 0;
for ( int i = 1; i < db; i ++ )
{
  if ( tomb[i].y > tomb[lm].y ) lm = i;
}
```

4. Statisztikai inforációk megjelenítése

```
cout << "\nElemzés:\n";

cout << "\nLegmagasabb pont: " << tomb[lm].y << " m,
          időpontja: " << tomb[lm].t << " s\n";
cout << t << " másodpercben már biztosan becsapódott.";
cout << "\nLegtávolabbi pont: " << tomb[db-1].x << " m";
```

5. „Grafikon" készítése

Pontosabban karakterek segítségével fogjuk ábrázolni az ágyúgolyó mozgását. Ehhez először készítettünk egy saját függvényt, amely a megadott pozicióba helyezi a kurzort.

```
void gotoxy( int oszlop, int sor )
{
  COORD coord;
  coord.X = oszlop;
  coord.Y = sor;
  SetConsoleCursorPosition(GetStdHandle( STD_OUTPUT_HANDLE
  ),coord);
}
```

Ezután a tömb elemeit az x, y koordinátáknak megfelelően kiíratjuk a képernyőre:

```
for ( int i = 0; i < db; i ++ )
{
    x = tomb[i].x/3;
    y = tomb[i].y/2;
    gotoxy(x,y);
    if ( i == lm ) cout << "x";
    else cout << "o";
}
```

torzítás, hogy elférjen a konzolablakban!

jelezzük a maximális magasságot

Konzolablak adta lehetőségek (bal felső sarokban van az origó!) miatti problémák kiküszöbölése: tükrözés, eltolás

```
for ( int i = 0; i < db; i ++ )
{
    x = tomb[i].x/3;
    y = (50 + tomb[lm].y - tomb[i].y)/2;
    gotoxy(x,y);
    if ( i == lm ) cout << "x";
    else cout << "o";
}
```