

```

1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9:
10: #include <iostream>
11: #include <cmath>
12: #include <iomanip>
13: using namespace std;
14:
15: int main()
16: {
17:     const long double PI = 3.1415;
18:     const long double g = 9.806;
19:
20:     int n, x0, y0, vel, elev, rot;           //valores de entrada
21:     double tiempo_vuelo, h_max, alcance, xf, yf; //valores de calculo y salida
22:
23:     cout << "Este programa calcula el alcance de un proyectil: " << endl;
24:
25:     //Entrada de datos...
26:     cout << "Introduce las coordenadas del canion(x0,y0): ";
27:     cin >> x0 >> y0;
28:     cout << "Introduce la velocidad de lanzamiento: ";
29:     cin >> vel;                               //Expresado en m/s
30:     if(vel<0){
31:         cout << "Error: La elevacion debe ser positiva";
32:         exit(-1);
33:     }
34:     cout << "Introduce la elevacion del canion: ";
35:     cin >> elev;                               //Expresado en grados
36:     if(elev>=90){
37:         cout << "Error: La elevacion debe estar entre 0-90 grados";
38:         exit(-1);
39:     }
40:     cout << "Introduce la rotacion del canion: ";
41:     cin >> rot;                               //Angulo 0 (en grados) es igual al Norte(Eje
y)
42:     if(rot>180 || rot<=-180){
43:         n=rot/180;                             //truncamiento de decimales por ser variable
entera
44:         rot = rot - 360*n;
45:         cout << "El angulo de rotacion rectificado es: " << rot << endl;
46:     }
47:
48:     const double elev_deg =elev*PI/180;        //La funcion sin está en radianes, por lo qu
e hay pasarla a grados.
49:     const double rot_deg = rot*PI/180;        //***
50:
51:     //Cálculos...
52:     tiempo_vuelo = (2*vel*sin(elev_deg))/g;
53:     h_max = ((vel*sin(elev_deg))*(vel*sin(elev_deg)))/(2*g);
54:     alcance = (vel*vel*sin(2*elev_deg))/g;      //Se repite vel a proposito para no tener qu
e usar pow(más tiempo de calculo)
55:     xf = x0 + alcance*sin(rot_deg);
56:     yf = y0 + alcance*cos(rot_deg);
57:
58:     //Salida de datos...
59:
60:     cout << "-----Resultado de Disparo-----"
61:         << "\nTiempo de vuelo:" << tiempo_vuelo << " s" << endl
62:         << "Altura maxima:" << h_max << " metros" << endl
63:         << "Alcance:" << alcance << " metros" << endl
64:         << "Impacto:" << xf << ", " << yf << " metros" << endl;
65:
66:     //-----Exito de Disparo-----
-----
67:
68:     //Entrada de datos...
69:
70:     cout << "-----Comprobacion de Objetivo-----";
71:
72:     double x, y, d, d0, d1;
73:
74:     cout << "\nIntroduzca la posicion del objetivo(x,y): ";
75:     cin >> x >> y;
76:     cout << "Introduzca distancias de impacto(dos valores): ";
77:     cin >> d0 >> d1;
78:     if(d0<0 || d1<0){
79:         cout << "Error: Las distancias de impacto no deben ser negativas";
80:         exit(-1);
81:     }

```

```
82:
83: //calcula de datos...
84:
85:     d = sqrt((x-xf)*(x-xf) + (y-yf)*(y-yf));           //Se repiten expresiones a proposito para no ten
er que usar pow(más tiempo de calculo)
86:
87: //Salida de datos...
88:
89:     cout << "La distancia al objetivo es: " << d;
90:     if(d<d0 && d<d1)
91:         cout << "---->Impacto Directo";
92:     else if(d>d0 && d>d1)
93:         cout << "---->No hay Impacto";
94:     else
95:         cout << "---->Impacto Parcial";
96:     return 0;
97: }
```