Péndulo

calcular max θ
dado P ~~(precisión / P > Er)~~

$$Er = \frac{T - T_{aprox}}{T_{aprox}} = \underbrace{\frac{1^2}{2^2}\,\mathrm{sen}^2\left(\frac{\theta}{2}\right)}_{i=1} + \underbrace{\frac{1^2\,3^2}{2^2\,4^2}\,\mathrm{sen}^4\left(\frac{\theta}{2}\right)}_{i=2} + \underbrace{\frac{1^2\,3^2\,5^2}{2^2\,4^2\,6^2}\,\mathrm{sen}^6\left(\frac{\theta}{2}\right)}_{i=3}$$

$n = 3$

? seguro ?

$$T_{aprox} = 2\pi\sqrt{\frac{L}{g}}$$

Er_{max} : 0,001 (0,1%)
Precisión : 1,0 (grados)

$$\sum \frac{(\text{impares} \leq n)^2}{(\text{pares} \leq 2n)^2} \cdot \left[\mathrm{sen}\left(\frac{\theta}{2}\right)\right]^{2n}$$

~~// número términos~~ ~~número términos aprox~~
~~suma = 0~~
~~for (int i=0, i<n, i++)~~
~~suma = suma~~ ~~if (i%2) // i impar~~
~~if (i%2)~~ ~~suma = suma~~
~~else // i par~~
~~else~~

#define maxn 100 ; nº términos de la aproximación

Calcular Er para n términos

```
Er = 0
for (int i=1; i≤n , i++)
{
    sumando = 1;
    for (int j=1; j≤i; j++)
    {
        if (j%2) // j impar
            sumando = sumando * j^2;
        else // j par
            sumando = sumando / j^2;
    }
    sumando = sumando * [sin(θ/2)]^2*i
    Er = Er + sumando;
}
```

unidades
decenas
centenas
milésimas
...

solicitar P // precisión de θ
esto no es eficiente, ya que debería ir de unidades a esa precisión conforme $Er(\theta) < Er_{max}$
solicitar Er_{max}
calcular $\theta_{(P)}$ / $Er(\theta) < Er_{max}$
seguramente se deba hacer con while

```
for ( fi ; Er(θ)<Ermax ; fi=fi+P)
{
~~for (int Pi=1; Pi ≥ P ; Pi = Pi/10)~~
for ( fi ; Er(θ) < Ermax ; fi += 1*Pi)
```

esto devuelve la θ max, y es el punto de partida para ir calculando la nueva con más decimales hasta llegar a la precisión

⋮

Acabar.