Primer TP modelos fisicos

Ejercicio 1

Para poder resolver de una forma "sencilla" vamos a pensar al problema como uno de vectores. Sabemos que si el origen de coordenadas es A entonces el vector P puede calcularse de la siguiente forma: (r es la longitud de la varilla AP)

$$\overline{P_A(t)} \ = \ (r \times \cos(\phi(t)), r \times \sin(\phi(t)))$$

El problema es que nos interesa saber el angulo que se forma entre el vector $\bar{i}$ cuando el origen de coordenadas se encuentra en B. Sin embargo, punto B es equivalente a $A + (0, l)$ siendo l la distancia entre ellos. De hecho, culaquier vector con origen de coordenadas B podemos trasladarlo a A sumando $\bar{l} = (0, l)$ o trasladar de A a B restando $\bar{l} = (0, l)$.

Entonces tenemos que en B:

$$\overline{P_B(t)} \ = \overline{P_A(t)} - \bar{l}$$

$$\overline{P_B(t)} \ = \ (r \times \cos(\phi(t)), r \times \sin(\phi(t))) - \bar{l}$$

$$= (r \times \cos(\phi(t)), r \times \sin(\phi(t))) - (0, l)$$

$$= (r \times \cos(\phi(t)), (r \times \sin(\phi(t))) - l)$$

Sabiendo esto podemos calcular:

$$\cos(\widehat{\bar{i}\overline{P_B(t)}}) = \frac{\bar{i} \times \overline{P_B(t)}}{|\bar{i}| \times |\overline{P_B(t)}|}$$

$$= \frac{r \times \cos(\phi(t))}{1 \times |\overline{P_B(t)}|}$$

$$= \frac{r \times \cos(\phi(t))}{\sqrt{(r \times \cos(\phi(t)))^2 \ + \ (r \times \sin(\phi(t)) - l)^2}}$$

$$= \frac{r \times \cos(\phi(t))}{\sqrt{r^2 \times \cos(\phi(t))^2 \ + \ (r \times \sin(\phi(t)) - l)^2}}$$

$$= \frac{r \times \cos(\phi(t))}{\sqrt{r^2 - 2 \times r \times \sin(\phi(t)) + l^2}}$$

Y por lo tanto:

$$\widehat{\bar{i}\overline{P_B(t)} = \cos^{-1}(\frac{r \times \cos(\phi(t))}{\sqrt{r^2 - 2 \times r \times \sin(\phi(t)) \times l + l^2}})}$$

Por simplicidad de la notación definimos:

$$f(t) = \cos^{-1}(\frac{r \times \cos(\phi(t))}{\sqrt{r^2 - 2 \times r \times \sin(\phi(t)) \times l + l^2}})$$

Pero esta no es la solución correcta. Esto nos da el menor ángulo entre $\bar{i}$ y $\overline{P_B(t)}$ por lo que hay que tener en cuenta si este angulo es el que buscamos.
Para determinar esto, simplemente hay que determinar si el punto P se encuentra en los cuadrantes 1 o 2. Eso lo podemos determinar si la segunda componente de $\overline{P_B(t)}$ es mayor a 0.

Simplificando:

$$\sin(\phi(t)) \geq \frac{l}{r}$$

Por lo tanto nos queda:

$$\theta(t) = \begin{cases} f(t), & \text{si } \sin(\phi(t)) \geq \frac{l}{r} \\ 2\pi - f(t), & \text{si } \sin(\phi(t)) < \frac{l}{r} \end{cases}$$

Y también:

$$\dot{\theta}(t) = \begin{cases} \dot{f}(t), & \text{si } \sin(\phi(t)) \geq \frac{l}{r} \\ -\dot{f}(t), & \text{si } \sin(\phi(t)) < \frac{l}{r} \end{cases}$$

Link al gráfico en geogebra: gráfico de mecanismo de retorno rápido de Whitworth

Ejercicio 2

La pelota al dejarse caer por los escalones describe una serie de parábolas que podemos calcular de esta manera:

$$R(t) = \begin{cases} x(t) & = x_0 + v_0 * seno(\alpha) * t \\ y(t) & = y_0 + e * v_0 * cos(\alpha) * t - (1/2) * g * t^2 \end{cases}$$

Siendo “e” el coeficiente de restitución de la pelota y fijando el centro de nuestro eje al inicio del primer escalón.

De esta manera, la altura inicial y_0 siempre es 0.

Para determinar si la pelota cae en el escalón en el que estamos “parados” obtenemos el tiempo que tarda en caer hasta la altura del escalón y revisamos si no superó el ancho del escalón.

$$tiempo = max(\frac{-velocidad \pm \sqrt{velocidad^2 - 4 * (-g/2) * -altura}}{2 * (-g/2)})$$

Si encontramos que la pelota rebota en el mismo escalón, mantenemos el eje, pero al momento de superarlo y caer cambiamos nuestro eje por el del nuevo escalón, modificando x_0 para contemplar el cambio de eje, y calculamos dónde rebota en el mismo.

De esta manera, se repite el procedimiento para determinar si la pelota rebota en el escalón en el que estamos “parados” hasta haber superado al último escalón.

Estos son los resultados obtenidos para los distintos casos de prueba especificados en el ejercicio

TEST: velocidad inicial=1.80 m/s; angulo=18.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 2 3 2 3 3 2 3
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 2 al menos dos veces

TEST: velocidad inicial=1.80 m/s; angulo=18.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 3 2 3 3 2 3 3
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 2 al menos dos veces

TEST: velocidad inicial=1.80 m/s; angulo=22.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 2 2 2 2 2 2 2
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 2 al menos dos veces

TEST: velocidad inicial=1.80 m/s; angulo=22.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 2 2 2 2 2 2 3
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 2 al menos dos veces

TEST: velocidad inicial=1.80 m/s; angulo=26.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 1 2 1 2 2 2 1
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 3 al menos dos veces

TEST: velocidad inicial=1.80 m/s; angulo=26.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 2 1 2 2 1 2 2
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 2 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=18.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 1 2 1 2 2 2 2
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 3 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=18.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 1 2 2 2 2 2 1
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 3 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=22.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 1 1 1 1 2 1 2
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 6 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=22.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 1 1 2 1 1 2 1
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 4 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=26.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 1 0 2 1 1 1 1
a) La pelota se salta un escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 4 al menos dos veces

TEST: velocidad inicial=2.40 m/s; angulo=26.00 grados; restitucion=0.5

cantidad de rebotes (escalon 1 a 8): 1 1 1 1 1 1 1 1
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota no rebota dos veces en un mismo escalon
c) La pelota no rebota dos veces en un mismo escalon

TEST: velocidad inicial=3.00 m/s; angulo=18.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 1 0 2 1 1 2 1
a) La pelota se salta un escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 4 al menos dos veces

TEST: velocidad inicial=3.00 m/s; angulo=18.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 1 1 1 1 2 1 2
a) La pelota baja por las escaleras sin saltarse ningun escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 6 al menos dos veces

TEST: velocidad inicial=3.00 m/s; angulo=22.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 0 1 1 0 2 1 1
a) La pelota se salta un escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 6 al menos dos veces

TEST: velocidad inicial=3.00 m/s; angulo=22.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 0 1 1 1 1 1 2
a) La pelota se salta un escalon
b) La pelota rebota dos veces sobre un mismo escalon
c) La pelota rebota sobre el escalon 8 al menos dos veces

TEST: velocidad inicial=3.00 m/s; angulo=26.00 grados; restitucion=0.6
cantidad de rebotes (escalon 1 a 8): 1 0 1 0 1 1 0 1
a) La pelota se salta un escalon
b) La pelota no rebota dos veces en un mismo escalon
c) La pelota no rebota dos veces en un mismo escalon

TEST: velocidad inicial=3.00 m/s; angulo=26.00 grados; restitucion=0.5
cantidad de rebotes (escalon 1 a 8): 1 0 1 1 0 1 1 1
a) La pelota se salta un escalon
b) La pelota no rebota dos veces en un mismo escalon
c) La pelota no rebota dos veces en un mismo escalon

Código con el que se realizaron los test:

```c
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <errno.h>

#include <math.h>

#define ANCHO_ESCALON 0.3 // m
#define ALTURA_ESCALON 0.15 // m

double get_double(const char* input){
  char *endptr = NULL;
  errno = 0;

  double ret = strtod(input, &endptr);

  if ((errno == ERANGE && (ret == LONG_MAX || ret == LONG_MIN)) || (errno != 0 && ret == 0)) {
```

```
    perror("strtod");
    exit(EXIT_FAILURE);
  }

  if (endptr == input) {
    fprintf(stderr, "No se encontraron digitos\n");
    exit(EXIT_FAILURE);
  }

  return ret;
}

void usage(char* progName){
  printf("%s\n", progName);
  printf("%s <velocidad_inicial: double (m/s)> <angulo: double (grados)> <reduccion_velocidad: double (%%)
  printf("Si no se ingresan los valores de velocidad inicial, ");
  printf("angulo y reduccion de velocidad, se utilizan valores de ");
  printf("velocidad desde 1.8 m/s hasta 3.0 m/s con incrementos ");
  printf("de 0.6 m/s, valores de angulo desde 18 grados hasta ");
  printf("26 grados con incrementos de 4 grados, y reduccion ");
  printf("de velocidad desde 40%% y 50%%\n\n");
}

double max(const double a, const double b){
  return (a > b) ? a : b;
}

// y(t) = v0*cos(ang)*t - 1/2 g t^2
// 0 = -g/2 * t^2 + v0*cos(ang) * t - y
// t1,t2
// tiempo = max(t1,t2)
double resolver_tiempo(double const altura, double const velocidad){
  double tiempo = 0;
  double const a = 4.905, dosA = -9.81, cuatroAC = 4*(-a)*(-altura);

  double t1 = (-velocidad + sqrt(pow(velocidad, 2) - cuatroAC)) / dosA;
  double t2 = (-velocidad - sqrt(pow(velocidad, 2) - cuatroAC)) / dosA;

  tiempo = max(t1,t2);

  return tiempo;
}

// Desparametrizamos las ecuaciones del tiro parabolico
// x(t) = x0 + v0*seno(ang)*t
// y(t) = y0 + v0*cos(ang)*t - 1/2 g t^2
//
// y(t) = y0 + t (v0*cos(ang) - g/2 t)
// y(t) = 0 => t = v0*cos(ang) / (g/2)
//
// x-x0 / v0*seno(ang) = t
// y(x) = cot(ang)*(x-x0) - g/2 * (x-x0 / v0*seno(ang))^2
// y(d) = cot(ang)*(d-x0) - (g/2) * (d-x0)^2 / (v*seno(ang))^2
// y(d) = 0 => cot(ang)*(d-x0) = (g/2) * (d-x0)^2 / (v*seno(ang))^2
// d = (v^2*seno(ang)*cos(ang) / (g/2)) + x0
double distancia_parabola(const double velocidad, const double sen_angulo, const double cos_angulo, const
```

```
  return ((pow(velocidad, 2.0) * sen_angulo * cos_angulo) / (4.9)) + posicion_inicial; // 4.9 == g/2
}

void test(double velocidad_inicial, double angulo, double restitucion){
  double posicion_inicial = 0.15;
  char *resultadoA="La pelota baja por las escaleras sin saltarse ningun escalon", *resultadoB="La pelota
  int escalon_doble_rebote = -1;

  printf("TEST: velocidad inicial=%.2lf m/s; angulo=%.2lf grados; restitucion=%.1lf\n", velocidad_inicial,

  const double rad_angulo = (angulo * M_PI) / 180.0;
  const double coseno = cos(rad_angulo), seno = sin(rad_angulo);

  int rebotes[8] = {0};

  const double velocidad_horizontal = velocidad_inicial * seno;
  double velocidad_vertical = velocidad_inicial * coseno * restitucion;

  // asumimos un primer rebote en 0.15m.
  rebotes[0] = 1;

  double t, distancia=0, siguiente_posicion, diff_escalon=0;
  int escalon_previo=0;
  // reviso si rebota en el mismo escalon.
  for (int escalon = 0; escalon < 8;) {
    t = resolver_tiempo((diff_escalon+1)*(-ALTURA_ESCALON), velocidad_vertical);

    distancia = posicion_inicial + velocidad_horizontal*t;

    if (distancia <= ANCHO_ESCALON) {
      rebotes[escalon] += 1;
      velocidad_vertical *= restitucion;
      posicion_inicial = distancia;
      escalon_previo = escalon;
      diff_escalon = (escalon - escalon_previo);
      // calculo en una nueva parabola.
    } else {
      // No rebota en este escalon, chequeamos el siguiente
      escalon_previo = escalon;
      escalon += distancia / ANCHO_ESCALON;
      // movemos el origen al siguiente escalon
      diff_escalon = (escalon - escalon_previo);
      posicion_inicial = posicion_inicial - diff_escalon*ANCHO_ESCALON;
    }
  }

  printf("cantidad de rebotes (escalon 1 a 8): ");
  for (int i=0; i<8; i++) {
    printf("%d ", rebotes[i]);
  }
  printf("\n");

  for (int i=0; i<8; i++) {
    if (escalon_doble_rebote == -1 && rebotes[i] >= 2) {
      escalon_doble_rebote = i;
      resultadoB = "La pelota rebota dos veces sobre un mismo escalon";
```

```
    }
    if (rebotes[i] == 0) {
      resultadoA = "La pelota se salta un escalon";
    }
  }

  printf("a) %s\n", resultadoA);
  printf("b) %s\n", resultadoB);
  if (escalon_doble_rebote != -1) {
    resultadoC = "La pelota rebota sobre el escalon";
    printf("c) %s %d al menos dos veces\n\n", resultadoC, escalon_doble_rebote+1);
  } else {
    printf("c) %s\n\n", resultadoC);
  }
}

int main(int argc, char *argv[])
{
  double velocidad_inicial; // en m/s
  double angulo; // en grados
  double perdida; // en %

  switch (argc) {
    case 1:
      for (velocidad_inicial = 1.8; velocidad_inicial <= 3.0; velocidad_inicial += 0.6) {
        for (angulo = 18; angulo <= 26; angulo += 4) {
          for (perdida = 40; perdida <= 50; perdida += 10) {
            test(velocidad_inicial, angulo, 1-(perdida/100));
          }
        }
      }
      break;
    case 4:
      velocidad_inicial = get_double(argv[1]);
      angulo = get_double(argv[2]);
      if (angulo > 90 || angulo < 0){
        fprintf(stderr, "El angulo en que se suelta no puede ser mayor a 90 grados o menor a 0 grados\n");
        exit(EXIT_FAILURE);
      }
      perdida = get_double(argv[3]);

      test(velocidad_inicial, angulo, perdida);
      break;
    default:
      printf("Cantidad de argumentos invalida.\n");
      usage(argv[0]);
      break;
  }

  return EXIT_SUCCESS;
}
```

Ejercicio 3

Apartado a) 1.

En este caso sabemos que la aceleración es constante y el auto se mueve sobre una rampa recta, por ende el movimiento es un MRUV.

Del enunciado sacamos los siguientes datos:

- $v_0 = 60\ mi/h$ (velocidad inicial)
- $v_f = 0\ mi/h$ (velocidad final)
- $a(t) = cte = -10\ ft/s^2$ (aceleración)

El programa se encarga de convertir estos valores al sistema USI.

Queremos calcular el tiempo t_f requerido para que el automóvil quede en reposo y la distancia d que recorre sobre la rampa.

Tomamos como tiempo inicial $t_0 = 0\ s$ y la posición inicial $x_0 = 0m$. Luego, la distancia recorrida será $d = x(t_f)$.

Como estamos en un MRUV, se cumple la siguiente ecuación: $v(t_f) = v_f = v_0 + a(t_f)(t_f - t_0)$.

Como $v_f = 0\ m/s$ y $t_0 = 0\ s$, resulta: $v_0 + a(t_f)t_f = 0\ m/s$.

Trabemos algebráicamente la expresión: $v_0 + a(t_f)t_f = 0\ m/s \Rightarrow a(t_f)t_f = -v_0 \Rightarrow t_f = -v_0/a(t_f)$.

Por lo tanto, obtuvimos

$$\mathbf{t_f = \frac{-v_0}{a(t_f)}}$$

Ahora queremos calcular $d = x(t_f)$. Por estar en un MRUV tenemos que se cumple esta ecuación: $x(t_f) = x_0 + v_0(t_f - t_0) + \frac{1}{2}a(t_f)(t_f - t_0)^2$.

Como $t_0 = 0\ s$ y $x_0 = 0\ m$, resulta: $x(t_f) = v_0 t_f + \frac{1}{2}a(t_f)t_f^2$.

Finalmente, como $d = x(t_f)$, obtuvimos

$$\mathbf{d = v_0\ t_f + \frac{1}{2}\ a(t_f)\ t_f^2}$$

Apartado a) 2.

En este caso tenemos que la aceleración varía linealmente, por lo cual su ecuación tiene la forma $a(t) = mt + a_0$, siendo $a_0 = -10\ ft/s^2$ y $a_f = a(t_f) = 0\ m/s^2$.

Nuevamente queremos calcular el tiempo t_f y la distancia d con las definiciones hechas en el apartado anterior.

Tenemos que $a(t) = \frac{dv}{dt} \Rightarrow dv = a(t)\ dt \Rightarrow \int_{v_0}^{v_f} dv = \int_{t_0}^{t_f} a(t)\ dt \Rightarrow \int_{v_0}^{v_f} dv = \int_{t_0}^{t_f} (mt + a_0)\ dt$

Como $t_0 = 0\ s$, resulta: $\int_{v_0}^{v_f} dv = m\int_0^{t_f} t\ dt + \int_0^{t_f} a_0\ dt \Rightarrow [v]_{v_0}^{v_f} = \frac{m}{2}\left[t^2\right]_0^{t_f} + a_0\,[t]_0^{t_f} \Rightarrow v_f - v_0 = \frac{1}{2}\ m\ t_f^2 + a_0\ t_f \Rightarrow v_f = v(t_f) = v_0 + \frac{1}{2}\ m\ t_f^2 + a_0\ t_f$

Como $v_f = 0\ m/s$, resulta: $-v_0 = \frac{1}{2}\ m\ t_f^2 + a_0\ t_f$

Luego podemos plantear el siguiente sistema de ecuaciones:

$$\begin{cases} mt_f + a_0 = a_f = 0\ m/s^2 \\ \frac{1}{2}\ m\ t_f^2 + a_0\ t_f = -v_0 \end{cases}$$

$$\Rightarrow$$

$$\begin{cases} mt_f = -a_0 \\ \frac{1}{2}\ (m\ t_f)\ t_f + a_0\ t_f = -v_0 \end{cases}$$

Obtenemos estas ecuaciones:

$$m t_f = -a_0 \tag{1}$$

$$\frac{1}{2}\,(m\ t_f)\ t_f + a_0\ t_f = -v_0 \tag{2}$$

Reemplazando (1) en (2) resulta: $\frac{1}{2}\ (-a_0)\ t_f + a_0\ t_f = -v_0 \Rightarrow a_0\ t_f - \frac{1}{2}\ a_0\ t_f = -v_0 \Rightarrow \frac{1}{2}\ a_0\ t_f = -v_0 \Rightarrow a_0\ t_f = -2v_0 \Rightarrow t_f = -2\frac{v_0}{a_0}$

Por lo tanto, obtuvimos que

$$\mathbf{t_f = -2\frac{v_0}{a_0}}$$

Y despejamos la pendiente $m = -\frac{a_0}{t_f}$.

Ahora nos queda calcular la distancia recorrida sobre la rampa. Para ello debemos obtener la ecuación del desplazamiento.

Partamos de que: $v(t) = \frac{dx}{dt} \Rightarrow dx = v(t)\ dt \Rightarrow \int_{x_0}^{x_f} dx = \int_{t_0}^{t_f} v(t)\ dt \Rightarrow \int_{x_0}^{x_f} dx = \int_{t_0}^{t_f} (v_0 + \frac{1}{2}\ m\ t^2 + a_0\ t)\ dt$

Como $t_0 = 0\ s$ y $x_0 = 0\ m$, resulta: $\int_0^{x_f} dx = v_0 \int_0^{t_f} dt + \frac{1}{2}\ m \int_0^{t_f} t^2\ dt + a_0 \int_0^{t_f} t\ dt \Rightarrow [x]_0^{x_f} = v_0\,[t]_0^{t_f} + \frac{m}{6}\,\left[t^3\right]_0^{t_f} + \frac{a_0}{2}\,\left[t^2\right]_0^{t_f} \Rightarrow x_f = v_0\ t_f + \frac{a_0}{2} t_f^2 + \frac{m}{6}\ t_f^3$

Por lo tanto, como $d = x_f = x(t_f)$, obtuvimos:

$$\mathbf{d = v_0\ t_f + \frac{a_0}{2} t_f^2 + \frac{m}{6}\ t_f^3}$$

Código con el que se resolvió el ejercicio:

```c
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define FT_TO_M 0.3048 // valor de 1 pie en metros
#define MI_TO_M 1609.34 // valor de 1 milla en metros
#define H_TO_S 3600 // valor de 1 hora en segundos

void usage();
void apartado_a1();
void apartado_a2();

int main(int argc, char *argv[]) {
  if (argc == 2) {
    int opc = atoi(argv[1]);
    switch (opc) {
      case 1: apartado_a1(); break;
      case 2: apartado_a2(); break;
      default: usage(); break;
    }
  }
  else {
    usage();
  }
}

void usage(){
  printf("OPCIONES:\n");
  printf("1: la rampa es recta y la aceleración permanece constante\n");
  printf("2: la rampa es recta y la aceleración varía linealmente\n");
```

```
}

void apartado_a1() {
  float a = -10 * FT_TO_M; // aceleración en m/s^2
  float v_0 = 60 * MI_TO_M / H_TO_S; // velocidad inicial en m/s

  float t_f = -v_0/a; // tiempo final en s
  printf("El tiempo requerido para que el automóvil quede en reposo con la aceleración constante dada es %

  float d = v_0 * t_f + a * pow(t_f, 2) / 2; // distancia recorrida en m, como x_0 = 0m, d es igual al val
  printf("La distancia recorrida sobre la rampa es %.2f m\n", d);
}

void apartado_a2() {
  float a_0 = -10 * FT_TO_M; // aceleración en m/s^2
  float v_0 = 60 * MI_TO_M / H_TO_S; // velocidad inicial en m/s

  float t_f = -2 * v_0 / a_0; // tiempo final en s
  printf("El tiempo requerido para que el automóvil quede en reposo con la aceleración variando linealment

  float m = - a_0 / t_f; // pendiente de la ecuación a(t) = mt + a_0
  float d = v_0 * t_f + a_0 * pow(t_f, 2) / 2 + m * pow(t_f, 3) / 6;
  printf("La distancia recorrida sobre la rampa es %.2f m\n", d);
}
```