Anahí Lizbeth Arceta Popoca
A01704959

Algoritmo: Cuestionario de Física

E0 = (1,2,3,4,5,6,7)

GRAVEDAD = 9.81

Listas

```
respuestas_usuario =[]
respuestas_programa = []
```

Limpiar listas

```
def limpiar_lista (respuestas_usuario, respuestas_programa, tam):
    del respuestas_usuario [0:tam]
    del respuestas_programa [0:tam]
```

Calcula la calificación del usuario

```
def calificacion (cont_f):
    tam = len(respuestas_programa)
    calif = (cont_f*100/tam)
    print ("\nTu calificación final es:", ("{:.2f}".format(calif)), "\n")
    limpiar_lista (respuestas_usuario, respuestas_programa, tam)
```

Correcto o Incorrecto

```
def correcto_o_incorrecto (respuestas_usuario, respuestas_programa, n_indice):
    contar = 0
    if (respuestas_usuario[n_indice]) == (respuestas_programa[n_indice]):
        contar = contar + 1
        print ("\nCorrecto")
        return contar
```

```
    else:
        print ("\nIncorrecto")
        return contar
```

Conversion_de_unidades

```
def kh_a_ms (kilometros_hora):
    metros_seg = (kilometros_hora*1000/3600)
    return ("{:.2f}".format(metros_seg))

def min_a_hrs (minutos):
    horas_c = minutos/60
    return horas_c
```

Tema 1_Operaciones_Notación científica

```
def notacion_cientifica(numero_nd):
    """Referencia: https://www.delftstack.com/es/howto/python/scientific-notation-python/"""
    return format(numero_nd, ".1E")

def notacion_decimal(numero_nc):
    """Referencia: https://www.it-swarm-es.com/es/python/convertir-notacion-cientifica-decimales/1051995017/"""
    return float(numero_nc)
```

Tema 1_Preguntas_Notación científica

```
def tema1_dificultad1(tema_e, dificultad_e):
    print ("\nTema 1: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t1n1_1 = input("""
    ¿Para qué se utiliza la notación científica?
```

Anahí Lizbeth Arceta Popoca
A01704959

```
a) Para expresar cantidades muy grandes o pequeñas
b) Solo para expresar cantidades muy grandes
c) Solo para expresar cantidades muy pequeñas
""")

respuestas_usuario.append(respuesta_t1n1_1)

respuesta_t1n1_1_p = "a"
respuestas_programa. append(respuesta_t1n1_1_p)

cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)



respuesta_t1n1_2 = input("""

¿Cuál es la notación científica de 3890000?
favor de expresar el valor como en el siguiente ejemplo:

1400 sería 1.4E+03

→ Solo un decimal
""")

respuestas_usuario.append(respuesta_t1n1_2)

respuesta_t1n1_2_p =(notacion_cientifica(3890000))
respuestas_programa. append(respuesta_t1n1_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)
```

```
    respuesta_t1n1_3 = float(input("""
    Para el número en notación científica a notación decimal:

    1.7E+05
    """))

    respuestas_usuario.append(respuesta_t1n1_3)

    respuesta_t1n1_3_p = (notacion_decimal ("1.7E+05"))
    respuestas_programa. append(respuesta_t1n1_3_p)

    cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)

    cont_f = cont_1 + cont_2 + cont_3

    calificacion (cont_f)



def tema1_dificultad2(tema_e, dificultad_e):
    print ("\nTema 1: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t1n2_1 = input("""
    Para convertir un número muy grande a notación científica,
    el punto decimal se moverá hacia la:

    a) Derecha
    b) Izquierda
    """)
```

```
respuestas_usuario.append(respuesta_t1n2_1)

respuesta_t1n2_1_p = "b"
respuestas_programa. append(respuesta_t1n2_1_p)

cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)


respuesta_t1n2_2 = input("""
En la notación científica el número:

a) Es multiplicado por un número que es elevado a la potencia 10
b) Es multiplicado por una potencia base 10 exponente entero
c) Es multiplicado por una potencia 10 elevada a un número negativo
""")

respuestas_usuario.append(respuesta_t1n2_2)

respuesta_t1n2_2_p = "b"
respuestas_programa. append(respuesta_t1n2_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)


respuesta_t1n2_3 = input("""

Convierte el número 0.000793807474 a notación científica:
Favor de expresar la respuesta de la siguiente manera:
```

```
a.bE±exponente
""")


respuestas_usuario.append(respuesta_t1n2_3)

respuesta_t1n2_3_p = (notacion_cientifica(0.000793807474))
respuestas_programa. append(respuesta_t1n2_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)



respuesta_t1n2_4 = input("""

Presenta en notación científica el resultado de la siguiente operación:

23.08/23.89 * (0.0000089 * 3)
""")

respuestas_usuario.append(respuesta_t1n2_4)

operacion_ad_1 = 23.08/23.89 * (0.0000089 * 3)

respuesta_t1n2_4_p = (notacion_cientifica (operacion_ad_1))
respuestas_programa. append(respuesta_t1n2_4_p)

cont_4 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 3)

cont_f = cont_1 + cont_2 + cont_3 + cont_4
```

```
    calificacion (cont_f)
```

Tema 2_Operaciones

```
def velocidad_mru(distancia_mru, tiempo_mru):
    velocidad_mru_f1 = distancia_mru/tiempo_mru
    return ("{:.2f}".format(velocidad_mru_f1))

def tiempo_mru(distancia_mru, velocidad_mru):
    velocidad_mru_f3 = distancia_mru/velocidad_mru
    return  ("{:.2f}".format(velocidad_mru_f3))
```

Distancia

```
def distancia_mrua_vfvit (velocidad_f_mrua, velocidad_i_mrua, tiempo_mrua):
    distancia_mrua_f2 = ((velocidad_f_mrua+velocidad_i_mrua)/2) * tiempo_mrua
    return ("{:.2f}".format(distancia_mrua_f2))

def distancia_mrua_vfvia (velocidad_f_mrua, velocidad_i_mrua,aceleracion_mrua):
    distancia_mrua_f3_p1 = (velocidad_f_mrua**2-velocidad_i_mrua**2)
    distancia_mrua_f3_p2 = distancia_mrua_f3_p1 / (2*aceleracion_mrua)
    return ("{:.2f}".format(distancia_mrua_f3_p2))
```

Velocidad Final

```
def velocidad_f_mrua_viat (velocidad_i_mrua, aceleracion_mrua, tiempo_mrua):
    velocidad_f_mrua_f1 = aceleracion_mrua*tiempo_mrua + velocidad_i_mrua
    return ("{:.2f}".format(velocidad_f_mrua_f1))
```

Aceleración

```
def aceleracion_mrua_vfvit (velocidad_f_mrua, velocidad_i_mrua, tiempo_mrua):
```

```
    aceleracion_mrua_f1 = (velocidad_f_mrua-velocidad_i_mrua)/tiempo_mrua
    return ("{:.2f}".format(aceleracion_mrua_f1))
```

Tema 2_Preguntas_MRU y MRUA

```
def tema2_dificultad1(tema_e, dificultad_e):
    print ("\nTema 2: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t2n1_1 = input("""

    ¿Qué significa movimiento uniforme?

    a) Cuando un objeto viaja a lo largo de una trayectoria,
       con aceleración nula y una velocidad constante
    b) Desplazamientos iguales que ocurren en intervalos sucesivos
       de tiempos iguales
    c) Cuando un objeto viaja a lo largo de una trayectoria, con
       aceleración constante y una magnitud de velocidad variable
    """)


    respuestas_usuario.append (respuesta_t2n1_1)

    respuesta_t2n1_1_p = "b"
    respuestas_programa.append(respuesta_t2n1_1_p)

    cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)


    respuesta_t2n1_2 = input("""
    ¿Cuál es la fórmula para calcular la velocidad en MRU?
```

```
a) v = d/t
b) v = d*t
c) v = d+t
""")

respuestas_usuario.append(respuesta_t2n1_2)

respuesta_t2n1_2_p = "a"
respuestas_programa.append(respuesta_t2n1_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)



respuesta_t2n1_3 = float(input("""
Calcula la velocidad de un auto que se mueve a 1450 m en 60 segundos
"""))

respuestas_usuario.append(str (respuesta_t2n1_3))

respuesta_t2n1_3_p = velocidad_mru(1450.0, 60.0)
respuestas_programa.append(respuesta_t2n1_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)



cont_f = cont_1 + cont_2 + cont_3

calificacion (cont_f)
```

Anahí Lizbeth Arceta Popoca
A01704959

```
def tema2_dificultad2(tema_e, dificultad_e):
    print ("\nTema 2: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t2n2_1 = input("""
    ¿Qué es necesario para que el movimiento de un objeto sea considerado
    Movimiento Rectilíneo Uniforme (MRU)?

    a) Que la aceleración pueda ser negativa o positiva
    b) Que la aceleración del objeto se constante
    c) Que la aceleración del objeto sea 0 m/s²
    """)

    respuestas_usuario.append(respuesta_t2n2_1)

    respuesta_t2n2_1_p = "c"
    respuestas_programa.append(respuesta_t2n2_1_p)

    cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)




    respuesta_t2n2_2 = float(input("""
    ¿Cuál es la aceleración de un objeto que de tener una velocidad de 0 m/s
    pasó a tener una velocidad de 55 m/s en 89 segundos?
    """))


    respuestas_usuario.append(str(respuesta_t2n2_2))
```

```
respuesta_t2n2_2_p = aceleracion_mrua_vfvit (55.0, 0.0, 89.0)
respuestas_programa.append(respuesta_t2n2_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)


respuesta_t2n2_3 = float(input("""
Calcula la distancia que recorre un objeto que inicias con una velocidad
de 3 km/h, termina con una velocidad de 45 m/s y tiene una aceleración de
4 m/s²
"""))

respuestas_usuario.append(str (respuesta_t2n2_3))

conversion_1 = float (kh_a_ms (3))

respuesta_t2n2_3_p = distancia_mrua_vfvia (45.0, conversion_1, 4.0)
respuestas_programa.append(respuesta_t2n2_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)


respuesta_t2n2_4 = float(input( """
Una grúa es utilizada para levantar una viga de acero de sección I hasta lo
alto de un edificio de 100 ft. Durante los primeros 2 s, la viga es
levantada del reposo con una aceleración hacia arriba de 8 ft/s2. Si la
velocidad permanece constante durante el resto del trayecto, ¿cuánto tiempo
se requiere en total para levantar la viga desde el suelo hasta el techo?
"""))
```

```
    respuestas_usuario.append(respuesta_t2n2_4)

    vf_t2n2_4 = float(velocidad_f_mrua_viat (0.0, 8, 2))
    d_t2n2_4 = float(distancia_mrua_vfvit (vf_t2n2_4, 0.0, 2))
    d_f_t2n2_4 = float((100-d_t2n2_4))

    tiempo_total_t2n2_4 = (float(tiempo_mru(d_f_t2n2_4, vf_t2n2_4)))+2

    respuesta_t2n2_4_p = tiempo_total_t2n2_4
    respuestas_programa.append(respuesta_t2n2_4_p)

    cont_4 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 3)


    cont_f = cont_1 + cont_2 + cont_3 + cont_4

    calificacion (cont_f)
```

Tema 3_Operaciones_Caída Libre

Altura

```
def altura_cl_t (tiempo_cl):
    altura_cl_f1 =(1/2)*GRAVEDAD*tiempo_cl**2
    return ("{:.2f}".format(altura_cl_f1))

def altura_cl_vivft (velocidad_i_cl, velocidad_f_cl, tiempo_cl):
    altura_cl_f3 = ((velocidad_i_cl + velocidad_f_cl)/2)*tiempo_cl
```

```
    return ("{:.2f}".format(altura_cl_f3))


def altura_cl_vivf (velocidad_i_cl, velocidad_f_cl):
    altura_cl_f4 = (velocidad_f_cl**2+velocidad_i_cl**2)/(2*GRAVEDAD)
    return ("{:.2f}".format(altura_cl_f4))
```

Tiempo

```
def tiempo_cl_vivfa (velocidad_i_cl, velocidad_f_cl, altura_cl):
    tiempo_cl_f3 = altura_cl/((velocidad_i_cl+velocidad_f_cl)/2)
    return ("{:.2f}".format(tiempo_cl_f3))
```

Velocidad Inicial

```
def velocidad_i_cl_vft (velocidad_f_cl, tiempo_cl):
    velocidad_i_cl_f1 = velocidad_f_cl-GRAVEDAD*tiempo_cl
    return ("{:.2f}".format(velocidad_i_cl_f1))


def velocidad_i_cl_at (altura_cl, tiempo_cl):
    velocidad_i_cl_f3 = (altura_cl-(1/2)*GRAVEDAD*tiempo_cl**2)/tiempo_cl
    return ("{:.2f}".format(velocidad_i_cl_f3))
```

Velocidad Final

```
def velocidad_f_cl_vit (velocidad_i_cl, tiempo_cl):
    velocidad_f_cl_f1 = velocidad_i_cl+GRAVEDAD*tiempo_cl
    return ("{:.2f}".format(velocidad_f_cl_f1))


def velocidad_f_cl_via (velocidad_i_cl, altura_cl):
    velocidad_f_cl_f2 = (velocidad_i_cl**2+2*GRAVEDAD*altura_cl)**(1/2)
    return ("{:.2f}".format(velocidad_f_cl_f2))


def velocidad_f_cl_viat (velocidad_i_cl, altura_cl, tiempo_cl):
```

```
    velocidad_f_cl_f3 = ((2*altura_cl)/tiempo_cl)-velocidad_i_cl
    return ("{:.2f}".format(velocidad_f_cl_f3))
```

Tema 3_Preguntas_Caída Libre

```
def tema3_dificultad1(tema_e, dificultad_e):
    print ("\nTema 3: " + tema_e + "\n\nNivel " + dificultad_e)

    print ("\nGravedad", GRAVEDAD)

    respuesta_t3n1_1 = input("""
    ¿Cuál sería la mejor definición para caída libre?

    a) Un objeto que cae
    b) Cualquier objeto bajo la acción de la gravedad en un lugar donde
       la resistencia del aire es despreciable
    c) Un objeto en caída libre
    """)

    respuestas_usuario.append(respuesta_t3n1_1)

    respuesta_t3n1_1_p = "b"
    respuestas_programa.append(respuesta_t3n1_1_p)

    cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)


    respuesta_t3n1_2 = float(input("""
    Calcula la altura de un objeto que cae en un tiempo de 27 segundos
    [considera que tu velocidad inicial es 0 m/s y tu gravedad de 9.81 m/s²]
```

```
    """))

    respuestas_usuario.append(str(respuesta_t3n1_2))

    respuesta_t3n1_2_p = altura_cl_t (27.0)
    respuestas_programa.append(respuesta_t3n1_2_p)

    cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)



    respuesta_t3n1_3 = float(input("""
    ¿Cuál es la velocidad final de un objeto que recorre 450 m durante 20
    segundos y que tiene una velocidad inicial de 13 m/s?
    """))


    respuestas_usuario.append("{:.2f}".format(respuesta_t3n1_3))

    respuesta_t3n1_3_p = velocidad_f_cl_viat(13.0, 450.0, 20.0)
    respuestas_programa.append(respuesta_t3n1_3_p)

    cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)

    cont_f = cont_1 + cont_2 + cont_3

    calificacion (cont_f)


def tema3_dificultad2(tema_e, dificultad_e):
    print ("\nTema 3: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
print ("\nGravedad", GRAVEDAD)

respuesta_t3n2_1 = float(input( """
Una persona suelta una piedra desde una azotea de 45 m de altura.
¿Cuánto tiempo tardará en llegar al suelo?
"""))

respuestas_usuario.append(str(respuesta_t3n2_1))

vf_t3n2_1 = float(velocidad_f_cl_via (0.0, 45.0))
respuesta_t3n2_1_p = tiempo_cl_vivfa (0.0, vf_t3n2_1, 45.0)
respuestas_programa.append(respuesta_t3n2_1_p)

cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)



respuesta_t3n2_2 = float(input( """
Una piedra se arroja verticalmente hacia abajo desde un puente y 4s
después cae en el agua con una velocidad de 60 m/s. Calcula la velocidad
inicial de la piedra.
"""))

respuestas_usuario.append(str(respuesta_t3n2_2))

respuesta_t3n2_2_p = velocidad_i_cl_vft (60.0, 4)
respuestas_programa.append(respuesta_t3n2_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)
```

```
respuesta_t3n2_3 = int(input( """
Desde el balcón de un edificio se deja caer una manzana y llega a la
planta baja en 5s. Si cada piso mide 2.88m ¿Desde qué piso cayo la
manzana?

(Introduce un número entero y redondea hacía arriba)
"""))

respuestas_usuario.append(str(respuesta_t3n2_3))

vf_t3n2_3 = float(velocidad_f_cl_vit (0.0, 5.0))
h_t3n2_3 = float(altura_cl_vivft (0.0, vf_t3n2_3, 5.0))

n_piso = round(h_t3n2_3/2.88)

"""Referencia: https://micro.recursospython.com/recursos/como-redondear-un-numero-decimal.html"""

respuesta_t3n2_3_p = str(n_piso)
respuestas_programa.append(respuesta_t3n2_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)

respuesta_t3n2_4 = input( """
Un objeto en caída libre tarda 2s en recorrer los últimos 40m antes de
golpear el suelo. Considerando esto calcula la altura (en metros) desde
donde se soltó
""")

respuestas_usuario.append(str(respuesta_t3n2_4))
```

```
    vi_t3n2_4 = float(velocidad_i_cl_at (40, 2))
    h_t3n2_4 = float(altura_cl_vivf(vi_t3n2_4, 0.0))

    respuesta_t3n2_4_p = str(h_t3n2_4 + 40)
    respuestas_programa.append(respuesta_t3n2_4_p)

    cont_4 = correcto_o_incorrecto(respuestas_usuario, respuestas_programa, 3)



    cont_f = cont_1 + cont_2 + cont_3 + cont_4
    calificacion (cont_f)
```

Tema 4_Operaciones_Leyes de Newton

```
def fuerza_friccion (normal, uk):
    f_friccion = normal*uk
    return ("{:.2f}".format(f_friccion))

def peso_newton (masa_n):
    peso_newton_f = masa_n*GRAVEDAD
    return ("{:.2f}".format(peso_newton_f))

def fuerza_neta (masa_n, aceleracion_n):
    fuerza_neta_f = masa_n *aceleracion_n
    return ("{:.2f}".format(fuerza_neta_f))

def masa_2n (fuerza_n, aceleracion_n):
    masa_2n_f = fuerza_n/aceleracion_n
    return ("{:.2f}".format(masa_2n_f))
```

```
def aceleracion_2n (fuerza_n, masa_n):
    aceleracion_2n_f = fuerza_n/masa_n
    return ("{:.2f}".format(aceleracion_2n_f))
```

Tema 4_Preguntas_Leyes de Newton

```
def tema4_dificultad1(tema_e, dificultad_e):
    print ("\nTema 4: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t4n1_1 = input("""
    "A toda acción corresponde una reacción de igual magnitud,
    pero en sentido contrario"

    ¿De qué ley estamos hablando?

    a) 1ra Ley de Newton
    b) 2da Ley de Newton
    c) 3ra Ley de Newton
    """)

    respuestas_usuario.append(str(respuesta_t4n1_1))

    respuesta_t4n1_1_p = "c"
    respuestas_programa.append(respuesta_t4n1_1_p)

    cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)


    respuesta_t4n1_2 = float(input("""
    ¿Cuál es la Fuerza Neta de un objeto con una masa de 40 Kg y una
```

```
aceleración = 3.5 m/s^2?
"""))

respuestas_usuario.append("{:.2f}".format(respuesta_t4n1_2))

respuesta_t4n1_2_p = fuerza_neta (40.0,3.5)
respuestas_programa.append(respuesta_t4n1_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)



respuesta_t4n1_3 = input("""
¿Qué otro nombre se le da a la primera Ley de Newton?

a) Karma
b) Ley de la Inercia
c) Ley de los objetos estáticos
""")

respuestas_usuario.append(respuesta_t4n1_3)

respuesta_t4n1_3_p = "b"
respuestas_programa.append(respuesta_t4n1_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)

cont_f = cont_1 + cont_2 + cont_3

calificacion (cont_f)
```

Anahí Lizbeth Arceta Popoca
A01704959

```
def tema4_dificultad2(tema_e, dificultad_e):
    print ("\nTema 4: " + tema_e + "\n\nNivel " + dificultad_e)

    respuesta_t4n2_1 = input ( """
    Un bloque es arrastrado a lo largo de una mesa, considerando que la fricción
    ejercida entre la mesa y el bloque es despreciable ¿Cuántas y cuáles fuerzas
    son las que están actuando sobre el bloque?

    a) 3 → Peso (W), Normal (N) y la Fuerza Neta (F)
    b) 4 → Peso (W), Normal (N), Fuerza de Fricción (Fκ) y la Fuerza Neta (F)
    c) 2 → Peso (W) y Normal (N)
    """)

    respuestas_usuario.append(str(respuesta_t4n2_1))

    respuesta_t4n2_1_p = "a"
    respuestas_programa.append(respuesta_t4n2_1_p)

    cont_1 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 0)



    respuesta_t4n2_2 = input( """
    Un hombre adulto y un niño pequeño están parados uno frente al otro sobre
    hielo sin fricción. Juntan sus manos y el hombre empuja al niño de modo
    que se separan ¿Quién se aleja con mayor rapidez?

    a) El hombre
    b) El niño
```

```
c) Ambos se alejan a la misma rapidez
""")

respuestas_usuario.append(str(respuesta_t4n2_2))

respuesta_t4n2_2_p = "b"
respuestas_programa.append(respuesta_t4n2_2_p)

cont_2 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 1)



respuesta_t4n2_3 = float(input( """
Una fuerza horizontal de 100N arrastra un bloque de 8kg horizontalmente
a lo largo del suelo. Si el coeficiente de fricción cinética entre el
bloque y el suelo es de 0.2, encuéntrese la aceleración del bloque.
"""))

respuestas_usuario.append(str(respuesta_t4n2_3))

peso_t4n2_3 = float(peso_newton (8.0))
fk_t4n2_3 = float(fuerza_friccion(peso_t4n2_3, 0.2))
suma_F_t4n2_3 = 100-fk_t4n2_3

respuesta_t4n2_3_p = aceleracion_2n (suma_F_t4n2_3, 8.0)
respuestas_programa.append(respuesta_t4n2_3_p)

cont_3 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 2)



respuesta_t4n2_4 = float(input("""
```

```
    Un ascensor de 2000 lb es subido con una aceleración de 8 ft/s2.
    Encuéntrese la resistencia mínima a la ruptura que debe tener el cable
    que lo soporta.

    El valor de la gravedad es: 32 ft/s²
    """))

    respuestas_usuario.append(respuesta_t4n2_4)

    masa_t4n2_4 = float(masa_2n (2000.0, 32.0))
    f1 = float(fuerza_neta (masa_t4n2_4, 8.0))
    suma_F_t4n2_4 = f1+2000

    respuesta_t4n2_4_p = suma_F_t4n2_4
    respuestas_programa.append(respuesta_t4n2_4_p)

    cont_4 = correcto_o_incorrecto (respuestas_usuario, respuestas_programa, 3)



    cont_f = cont_1 + cont_2 + cont_3 + cont_4
    calificacion (cont_f)

Elegir tema y nivel
def tema_y_nivel ():

    tema = int(input("""
    1.- Notación científica
    2.- MRU y MRUA
    3.- Caída Libre
    4.- Leyes de Newton
```

```
"""))


if tema < 1 or tema > 5:
    tema = int(input("""
Lo siento el número seleccionado no está en los parámetros,
por favor ingrese un número que este dentro de los parámetros:

1.- Notación científica
2.- MRU y MRUA
3.- Caída Libre
4.- Leyes de Newton
"""))


dificultad = int(input("""
¿Qué nivel deseas completar? Escribe el número:

1.- Sencillo
2.- Normal
"""))

if dificultad < 0 or dificultad > 2:

    dificultad = int(input("""
Lo siento el número seleccionado no está en los parámetros,
por favor ingrese un número que este dentro de los parámetros:

1.- Sencillo
2.- Normal
```

```
"""))

print ("""Instrucciones:
En preguntas que requieran una respuesta de valor numérico, el valor
deberá de ingresarse siempre con dos decimales a excepción de que haya
una instrucción que señale lo contrario, esto también aplica para las
respuestas con valores enteros, en cuyo caso los decimales ingresados
deberán ser .00, al mismo tiempo NO deberán de ingresarse las unidades
de medición de estas respuestas.

Asimismo, las respuestas deberán de estar redondeadas hacia arriba
en caso de que tu tercer decimar sea igual o mayor a 5 o hacia abajo
en caso de que tu tercer decimar sea menor a 5, de no cumplir con estos
criterios tu respuesta se considerará incorrecta.
""")

if tema == 1 and dificultad == 1:
    tema1_dificultad1("Notación Cinetífica", "1")

elif tema == 1 and dificultad == 2:
    tema1_dificultad2("Notación Cinetífica", "2")




elif tema == 2 and dificultad == 1:
    tema2_dificultad1("MRU y MRUA", "1")

elif tema == 2 and dificultad == 2:
    tema2_dificultad2("MRU y MRUA", "2")
```

```
    elif tema == 3 and dificultad == 1:
        tema3_dificultad1("Caída Libre", "1")

    elif tema == 3 and dificultad == 2:
        tema3_dificultad2("Caída Libre", "2")



    elif tema == 4 and dificultad == 1:
        tema4_dificultad1("Leyes de Newton", "1")

    elif tema == 4 and dificultad == 2:
        tema4_dificultad2("Leyes de Newton", "2")
```

Algoritmo principal

```
print ("¡Hola usuario! ¿Qué tema deseas repasar? Por favor ingresa el número")
tema_y_nivel ()

continuar_p = input("""\n¿Deseas repasar otro tema?
a) Sí
b) No
""")



while continuar_p == "a" or continuar_p == "A":
    tema_y_nivel ()
    continuar_p = input("""
    ¿Deseas repasar otro tema?
    a) Sí
    b) No
```

Anahí Lizbeth Arceta Popoca
A01704959

"""")

EF ("\n¡Gracias por repasar en este programa!")

NOTA:

Aclaración nombre de variables

Respuesta_t1n1_1_p

t → Tema: indica el número de tema

n → Nivel: Indica el número de nivel

1 → Número de pregunta

p → Respuesta dada por el programa, en caso de que la *p* no aparezca indica que es la respuesta del usuario