

ALGORITMO: CUESTIONARIO DE FÍSICA

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

global nivel

global calificacionfinal

calificacionfinal = 0

global buscar

global velocidad_mru

global distancia_mru

global tiempo_mru

global velocidad_i_mrue

global velocidad_f_mrue

global tiempo_mrue

global distancia_mrue

global aceleración_mrue

global gravedad_cl

global gravedad_tv

global tiempo_cltv

global velocidad_i_cltv

global velocidad_f_cltv

global altura_cltv

global masa

global peso

global aceleracion_2ln

global fuerza

buscar = “ ”

velocidad_mru = 0.0

distancia_mru = 0.0

tiempo_mru = 0.0

velocidad_i_mrui = 0.0

velocidad_f_mrui = 0.0

tiempo_mrui = 0.0

distancia_mrui = 0.0

aceleración_mrui = 0.0

gravedad_cl = 9.81

gravedad_tv = -9.81

tiempo_cltv = 0.0

velocidad_i_cltv = 0.0

velocidad_f_cltv = 0.0

altura_cltv = 0.0

masa = 0.0

peso = 0.0

aceleracion_2ln = 0.0

fuerza = 0.0

```
tema = int(input("¡¡¡¡¡Hola usuario! Bienvenido a “(Nombre del proyecto)”, ¿Qué tema  
deseas repasar? Por favor ingresa el número:
```

- 1.- Notación científica
- 2.- MRU y MRUA
- 3.- Caída Libre y Tiro Vertical
- 4.- Masa y Peso
- 5.- Leyes de Newton))))))

mientras tema != (1,2,3,4,5,6,7):

```
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 y Tiro Vertical
- 4.- Masa y Peso
- 5.- Leyes de Newton''''''''))

si tema == 1:

llamar función nivel_seleccionado()

llamar función tema1 ()

```
si tema == 2:
```

llamar función nivel_seleccionado()

llamar función tema2 ()

```
si tema == 3:
```

llamar función nivel_seleccionado()

llamar función tema3 ()

si tema == 4:

llamar función nivel_seleccionado()

llamar función tema4 ()

si tema == 5:

llamar función nivel_seleccionado()

llamar función tema5 ()

función nivel_seleccionado ():

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

1.- Sencillo

2.- Normal

3.- Avanzado"))))

mientras nivel != (1,2,3):

nivel = 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

3.- Avanzado"))))

función tema1 ():

si nivel ==1:

respuesta1n1_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de
nivel/niveles)_(Número de pregunta/preguntas) ()

```
respuestat1n1_1 = float(input("Pregunta \n"))
```

Depende del problema
que se haya planteado

```
{ guardar_nnd = float ("Guardar el número decimal")  
  guardar_nnc_cadena = ("Guardar el número en notación científica como un  
  string")
```

llamar función notacion_cientifica_p () o notacion_cientifica_d () → Según corresponda

si nivel ==2:

```
respuestat1n2_1 = input("Pregunta \n")
```

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

```
respuestat1n2_1 = float(input("Pregunta \n"))
```

Depende del problema
que se haya planteado

```
{ guardar_nnd = float ("Guardar el número decimal")  
  guardar_nnc_cadena = ("Guardar el número en notación científica como un  
  string")
```

llamar función notacion_cientifica_p () o notacion_cientifica_d () → Según corresponda

si nivel ==3:

```
respuestat1n3_1 = input("Pregunta \n")
```

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

```
respuestat1n3_1 = float(input("Pregunta \n"))
```

Depende del problema
que se haya planteado

$\left\{ \begin{array}{l} \text{guardar_nnd} = \text{float}(\text{"Guardar el número decimal"}) \\ \text{guardar_nnc_cadena} = (\text{"Guardar el número en notación científica como un string"}) \end{array} \right.$

llamar función `notacion_cientifica_p()` o `notacion_cientifica_d()` → Según corresponda

función `tema2()`:

si `nivel == 1`:

`respuestat2n1_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas)()`

`respuestat2n1_2 = float(input("Pregunta \n"))`

`buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración")` ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar función `mru()` o función `mrva()` → Según corresponda

si `nivel == 2`:

`respuestat2n2_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas)()`

`respuestat2n2_2 = float(input("Pregunta \n"))`

`buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración")` ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar función `mru()` o función `mrva()` → Según corresponda

si nivel ==3:

respuestat2n3_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat2n3_2 = float(input("Pregunta \n"))

buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion mru () o funcion mrva () → Según corresponda

función tema3 ():

si nivel ==1:

respuestat3n1_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat3n1_2 = float(input("Pregunta \n"))

buscar = ("altura / tiempo / velocidad inicial / velocidad final") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida_libre () o funcion tiro_vertical () → Según corresponda

si nivel ==2:

respuestat3n2_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat3n2_2 = float(input("Pregunta \n"))

buscar = ("altura / tiempo / velocidad inicial / velocidad final") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida_libre () o funcion tiro_vertical () → Según corresponda

si nivel ==3:

respuestat3n3_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat3n3_2 = float(input("Pregunta \n"))

buscar = ("altura / tiempo / velocidad inicial / velocidad final") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida_libre () o funcion tiro_vertical () → Según corresponda

función tema4 ():

si nivel ==1:

respuestat4n1_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat4n1_2 = float(input("Pregunta \n"))

buscar = ("masa / peso") ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa_y_peso ()

si nivel ==2:

respuestat4n2_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat4n2_2 = float(input("Pregunta \n"))

buscar = ("masa / peso") ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa_y_peso ()

si nivel ==3:

respuestat4n3_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat4n3_2 = float(input("Pregunta \n"))

buscar = ("masa / peso") ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa_y_peso ()

función tema5 ():

si nivel ==1:

respuestat5n1_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat5n1_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o acleracion_2ln, según sea el caso

llamar función 2_ley_newton ()

si nivel ==2:

respuestat5n2_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat5n2_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o acleracion_2ln, según sea el caso

llamar función 2_ley_newton ()

si nivel ==3:

respuestat5n3_1 = input("Pregunta \n")

llamar función respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()

respuestat5n3_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o acleracion_2ln, según sea el caso

llamar función 2_ley_newton ()

Nota_1: Todas las variables marcadas al principio, ya que habrán sido modificada, en el mismo apartado en el que se hizo la pregunta, por el nuevo valor correspondiente necesario para la realizar la formula.

funcion notacion_cientifica_p ():

```
num_nc = format( guardar_nnd, “.1E”)
```

```
si respuesta(Número de tema)(Número de nivel) _(Número de pregunta) ==  
num_nc:
```

```
    imprimir (“Respuesta correcta, ¡Felicidades!”)
```

```
    calificacionfinal = calificacionfinal + int(Valor que tenga esa pregunta)
```

```
else:
```

```
    imprimir (“Respuesta incorrecta, la respuesta correcta sería”, Num_nc,  
    “”“”“porque:
```

```
    (Explicación corta del por qué es incorrecta) ”””””)
```

funcion notacion_cientifica_d ()

```
num_nd = float (guardar_nnc_cadena)
```

```
si respuesta(Número de tema)(Número de nivel) _(Número de pregunta) ==  
num_nd:
```

```
    imprimir (“Respuesta correcta, ¡Felicidades!”)
```

```
    calificacionfinal = calificacionfinal + int(Valor que tenga esa pregunta)
```

```
else:
```

```
    imprimir (“Respuesta incorrecta, la respuesta correcta sería”, Num_nc,  
    “”“”“porque:
```

(Explicación corta del por qué es incorrecta) """)

funcion mru ():

si distancia_mru > 0 and tiempo_mru > 0:

velocidad_mru = float (distancia_mru*tiempo_mru)

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_mru > 0 and distancia_mru == 0:

distancia_mru = float (velocidad_mru* tiempo_mru)

llamar funcion calificación_f ()

llamar funcion clear ()

si distancia_mru > 0 and tiempo_mru == 0:

tiempo_mru = float (distancia_mru / velocidad_mru)

llamar funcion calificación_f ()

llamar funcion clear ()

funcion mrua ():

si buscar == "distancia":

si aceleración_mrua != 0 and tiempo_mrua >= 0:

```
distancia_mrúa = (velocidad_i_mrúa * tiempo_mrúa + ( (1/2)*  
aceleración_mrúa* tiempo_mrúa**2))
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si tiempo_mrúa >= 0 and aceleración_mrúa ==0:
```

```
distancia_mrúa = (((velocidad_f_mrúa + velocidad_i_mrúa) / 2) *  
tiempo_mrúa)
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si aceleración_mrúa != 0 and tiempo_mrúa == 0:
```

```
distancia_mrúa = (velocidad_f_mrúa**2 - velocidad_i_mrúa**2) /  
(2*aceleración_mrúa)
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si buscar == “tiempo”:
```

```
si aceleración_mrúa != 0 and distancia_mrúa == 0:
```

```
tiempo_mrúa = (velocidad_f_mrúa - velocidad_i_mrúa) /  
aceleración_mrúa
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si distancia_mrúa >= 0 and aceleración_mrúa ==0:
```

```
tiempo_mrúa = (distancia_mrúa / ((velocidad_f_mrúa +  
velocidad_i_mrúa) / 2))
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

si aceleración_mrúa != 0 and distancia_mrúa >= 0 and velocidad_i_mrúa == 0:

tiempo_mrúa = ((2* distancia_mrúa) / aceleracion_mrúa)**(1/2)

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad inicial”:

si aceleración_mrúa != 0 and tiempo_mrúa >=0:

velocidad_i_mrúa = (velocidad_i_mrúa - (aceleración_mrúa*
tiempo_mrúa))

llamar funcion calificación_f ()

llamar funcion clear ()

si aceleracion_mrúa != 0 and distancia_mrúa >= 0:

velocidad_i_mrúa = velocidad_f_mrúa - 2 * distancia_mrúa *
aceleracion_mrúa

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad final”:

si aceleración_mrúa != 0 and tiempo_mrúa >= 0:

velocidad_f_mrúa = (velocidad_i_mrúa + (aceleración_mrúa*
tiempo_mrúa))

llamar funcion calificación_f ()

llamar funcion clear ()

si aceleración_mrúa != 0 and distancia_mrúa >= 0:

```
velocidad_f_mrúa = 2*distancia_mrúa* aceleracion_mrúa +  
velocidad_i_mrúa**2
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si buscar == “aceleración”:
```

```
si tiempo_mrúa >= 0 and distancia == 0:
```

```
aceleración_mrúa = (velocidad_f_mrúa - velocidad_i_mrúa) /  
tiempo_mrúa
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si distancia_mrúa >= 0 and tiempo_mrúa ==0:
```

```
aceleracion_mrúa = (velocidad_f_mrúa**2 - velocidad_i_mrúa**2) /  
(2*distancia_mrúa)
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
funcion caida_libre ():
```

```
si buscar == “altura”:
```

```
si tiempo_cltv >= 0 and velocidad_i_cltv != 0 or velocidad_f_cltv != 0:
```

```
altura_cltv = ((velocidad_i_cltv + or velocidad_f_cltv)/2) *  
tiempo_cltv
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si tiempo_cltv >= 0 and velocidad_i_cltv != 0 or velocidad_f_cltv == 0:
```

```
altura_cltv = (velocidad_i_cltv * tiempo_cltv) + (1/2) * gravedad_cl  
* tiempo_cltv**2
```

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= 0 and or velocidad_i_cltv == 0:

altura_cltv = $(1/2) * gravedad_cl * tiempo_cltv^{**2}$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0:

altura_cltv = $(velocidad_f_cltv - velocidad_i_cltv) / (2 * gravedad_cl)$

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “tiempo”:

si altura_cltv >= 0 and velocidad_i_cltv == 0:

tiempo_cltv = $((2 * altura_cltv) / gravedad_cl)^{** (1/2)}$

llamar funcion calificación_f ()

llamar funcion clear ()

si altura_cltv >= 0:

tiempo_cltv = $altura_cltv / ((velocidad_i_cltv + velocidad_f_cltv) / 2)$

llamar funcion calificación_f ()

llamar funcion clear ()

si velocidad_i_cltv != 0 or velocidad_f_cltv != 0:

$\text{tiempo_cltv} = (\text{velocidad_f_cltv} - \text{velocidad_i_cltv}) / \text{gravedad_cl}$

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad inicial”:

si tiempo_cltv >= 0 and altura_cltv == 0:

$\text{velocidad_i_cltv} = \text{velocidad_f_cltv} - \text{gravedad_cl} * \text{tiempo_cltv}$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0 and altura_cltv >= 0:

$\text{velocidad_i_cltv} = \sqrt{\text{velocidad_f_cltv}^2 - 2 * \text{gravedad_cl} * \text{altura_cltv}}$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= and velocidad_f_cltv != 0 and altura_cltv >= 0:

$\text{velocidad_i_cltv} = (\text{altura_cltv} - (1/2) * \text{gravedad_cl} * (\text{tiempo_cltv}^2)) / \text{tiempo_cltv}$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= and altura_cltv >= 0:

$\text{velocidad_i_cltv} = ((2 * \text{altura_cltv}) / \text{tiempo_cltv}) - \text{velocidad_f_cltv}$

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad final”:

si tiempo_cltv >= 0 and altura_cltv == 0:

velocidad_f_cltv = velocidad_i_cltv + gravedad_cl * tiempo_cltv

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0 and altura_cltv >= 0:

velocidad_f_cltv = velocidad_i_cltv **2 + 2 * gravedad_cl *
altura_cltv

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= and altura_cltv >= 0:

velocidad_f_cltv = ((2*altura_cltv) / tiempo_cltv) – velocidad_i_cltv

llamar funcion calificación_f ()

llamar funcion clear ()

funcion caida_libre ():

si buscar == “altura”:

si tiempo_cltv >= 0 and velocidad_i_cltv != 0 or velocidad_f_cltv == 0:

altura_cltv = (velocidad_i_cltv * tiempo_cltv) + (1/2) * gravedad_tv
* tiempo_cltv**2

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= 0 and or velocidad_i_cltv == 0:

altura_cltv = (1/2) * gravedad_tv * tiempo_cltv**2

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0:

altura_cltv = (velocidad_f_cltv – velocidad_i_cltv) / (2*gravedad_tv)

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “tiempo”:

si altura_cltv >= 0 and velocidad_i_cltv == 0:

tiempo_cltv = ((2*altura_cltv) / gravedad_tv)**(1/2)

llamar funcion calificación_f ()

llamar funcion clear ()

si altura_cltv >= 0:

tiempo_cltv = altura_cltv / ((velocidad_i_cltv + velocidad_f_cltv) / 2)

llamar funcion calificación_f ()

llamar funcion clear ()

si velocidad_i_cltv != 0 or velocidad_f_cltv != 0:

tiempo_cltv = (velocidad_f_cltv - velocidad_i_cltv) / gravedad_tv

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad inicial”:

si tiempo_cltv >= 0 and altura_cltv == 0:

$velocidad_i_cltv = velocidad_f_cltv - gravedad_tv * tiempo_cltv$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0 and altura_cltv >= 0:

$velocidad_i_cltv = velocidad_f_cltv^{**2} - 2 * gravedad_tv * altura_cltv$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= and velocidad_f_cltv != 0 and altura_cltv >= 0:

$velocidad_i_cltv = (altura_cltv - (1/2)*gravedad_tv * (tiempo_cltv^{**2})) / tiempo_cltv$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv >= and altura_cltv >= 0:

$velocidad_i_cltv = ((2*altura_cltv) / tiempo_cltv) - velocidad_f_cltv$

llamar funcion calificación_f ()

llamar funcion clear ()

si buscar == “velocidad final”:

si tiempo_cltv >= 0 and altura_cltv == 0:

$velocidad_f_cltv = velocidad_i_cltv + gravedad_tv * tiempo_cltv$

llamar funcion calificación_f ()

llamar funcion clear ()

si tiempo_cltv == 0 and altura_cltv >= 0:

```
velocidad_f_cltv = velocidad_i_cltv **2 + 2 * gravedad_tv *  
altura_cltv
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si tiempo_cltv >= and altura_cltv >= 0:
```

```
velocidad_f_cltv = ((2*altura_cltv) / tiempo_cltv) – velocidad_i_cltv
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
función masa_y_peso ():
```

```
si buscar == “masa”:
```

```
masa = peso / gravedad_cl
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
si buscar == “peso”:
```

```
peso = masa * gravedad_cl
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

```
función 2_ley_newton ():
```

```
si buscar == “fuerza”:
```

```
fuerza = masa * aceleracion_2ln
```

```
llamar funcion calificación_f ()
```

```
llamar funcion clear ()
```

si buscar == “aceleración”:

$aceleracion_2ln = fuerza / masa$

 llamar funcion calificación_f ()

 llamar funcion clear ()

si buscar == “masa”:

$masa = fuerza / aceleracion_2ln$

 llamar funcion calificación_f ()

 llamar funcion clear ()

funcion clear ():

 buscar = “ ”

$velocidad_mru = 0.0$

$distancia_mru = 0.0$

$tiempo_mru = 0.0$

$velocidad_i_mrua = 0.0$

$velocidad_f_mrua = 0.0$

$tiempo_mrua = 0.0$

$distancia_mrua = 0.0$

$aceleración_mrua = 0.0$

$tiempo_cltv = 0.0$

$velocidad_i_cltv = 0.0$

$velocidad_f_cltv = 0.0$

$altura_cltv = 0.0$

masa = 0.0

peso = 0.0

aceleracion_2ln = 0.0

fuerza = 0.0

funcion calificación_f ():

si respuesta(*Número de tema*)(*Número de nivel*) _(*Número de pregunta*) ==
(*Resultado de la operación Correspondiente*):

imprimir (“Respuesta correcta, ¡Felicidades!”)

calificacionfinal = calificacionfinal + int(*Valor que tenga esa pregunta*)

else:

imprimir (“Respuesta incorrecta, la respuesta correcta sería”, operacion,
“”“”“porque:

(Explicación corta del por qué es incorrecta) ””””)

EF = print (“¡Hola de nuevo usuario! Tu calificación en esta prueba es:”, calificacionfinal)

Nota_2: El nombre del proyecto está ampliamente sujeto al cambio

Las partes entre paréntesis y cursiva se modificarán de acuerdo con el nombre que reciba la función, dependiendo de que pregunta o preguntas responde, el tema y nivel en que están dichas preguntas

respuesta(*Número de tema*)(*Número de nivel*) _(*Número de pregunta/preguntas*) →
respuesta5n1_1