

PDF Parcial I-2.pdf



TEC

Tecnológico de Costa Rica

Prof. Milton Villegas Lemus

Introducción a la Programación

April 8, 2014

Examen Parcial 1

82

Nombre: Fabian A. Solano Madrid Carnet: 2014068784

1. Construya programa que de una hilera de números de 0 al 5, delimitada por el símbolo @ a la izquierda y a la derecha, halle si la cantidad de símbolos 4 que viene es divisible por 3 y la cantidad de símbolos 3 es divisible por 7. La función debe tener un módulo que valide el string de entrada como el alfabeto, dos módulos recursivos que construyan dos listas una con la cantidad correspondiente de símbolos 4 y otra con la cantidad de símbolos 3. La función debe arrojar claramente el mensaje si la cantidad de símbolos 4 es divisible por 3 y/o la cantidad de símbolos 3 es divisible por 7. También debe especificar cuando no se da ninguno de los dos casos. (30 pts)

Nombre de la función 4div3\_3div7('hilera')

1. 4div3\_3div7([2,3])  
'Error argumento debe ser tipo string' ✓
2. 4div3\_3div7('2312@123@')  
'Hilera no comienza con delimitador definido (@)' ✗
3. 4div3\_3div7('@0234567@')  
'La cantidad de dígitos 4 no es divisible por 3' ✗  
'La cantidad de dígitos 3 no es divisible por 7' ✗
4. 4div3\_3div7('@0323456370303335@')  
'La cantidad de dígitos 4 no es divisible por 3' ✗  
'La cantidad de dígitos 3 es divisible por 7' ✗
5. 4div3\_3div7('@403234563703033354@')  
'La cantidad de dígitos 4 es divisible por 3' ✗  
'La cantidad de dígitos 3 es divisible por 7' ✗
6. 4div3\_3div7('@032344563703054@')  
'La cantidad de dígitos 4 es divisible por 3' ✗  
'La cantidad de dígitos 3 no es divisible por 7' ✗
7. 4div3\_3div7('@@032344563703054')  
'Error la hilera no termina con el delimitador definido (@)' ✗
8. lista\_tres\_aux('@032344563703054@')  
[3,3,3] ✗
9. lista\_cuatro\_aux('@032344563703054@')  
[4,4,4] ✗
10. lista\_cuatro\_aux('@@05@')  
'Error delimitadores (@) en el intermedio de la hilera'

@443344@

2. Construya una función que puede recibir 4 parámetros o argumentos y valide tres de los argumentos debe devolver el cuarto argumento faltante. (20 pts.)

La función es  $v_f = v_i - (a * t)$

$v_i$  = Velocidad Inicial (Real, Entero, Positivo)

$v_f$  = Velocidad Final (Real, Entero, Positivo)

$a$  = Aceleración (Real, Entero)

$t$  = tiempo (Real positivo)



April 8, 2014

Nombre del programa vel\_ace\_t("v\_i", "v\_f", "a", "t"),

Invocaciones:

1. vel\_ace\_t("40.0", "", "9.8", "15.0")  
'Velocidad Inicial: 40.0'  
'Aceleracion: 9.8'  
'Tiempo: 15.0 '  
'Velocidad Final: 187.0'
2. vel\_ace\_t("40", "", "velfin", "9.8")
3. 'Error: 'velfin' no se puede convertir a número Real'
4. vel\_ace\_t("40", "20", "9.8", "10")
- 2 5. 'Error: debe existir una incógnita dentro de los parámetros'
6. vel\_ace\_t("40", "", "", "10")
7. 'Error: existe más de una incógnita dentro de los parámetros'
8. vel\_ace\_t("40", "", "9.8", "15")  
'Velocidad Inicial: 40'  
'Aceleracion: 9.8'  
'Tiempo: 15 '  
'Velocidad Final: 187.0'
9. vel\_ace\_t("40.0", "", "9.8", "-15.0")  
'Error: No definido para tiempo Negativo'
10. vel\_ace\_t("40", "", "-9.8", "15")  
'Velocidad Inicial: 40'  
'Aceleracion: 9.8'  
'Tiempo: 15 '  
'Velocidad Final: -107.0'
3. Construya una función que calcule el coseno de un ángulo en radianes, donde los valores de n estarán en el intervalos [10,20], de acuerdo con la siguiente fórmula.

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}, \forall x$$

- a. La función se debe invocar de la forma cos(X,N). Debe haber un módulo llamado facto(X) que devuelve el valor de la función factorial de X. Si el valor de n está fuera de intervalo se debe notificar. (40 pts)

1. Cos(2\*3.1415,20)  
0.9999949269133721
2. Cos(0,20)  
1
3. Cos(2\*3.1415,15)  
0.9999949267876308
4. Cos(0,20)  
1
5. Cos(3.1415,15)  
-0.9999987317275397
6. Cos(1.57075,20)  
4.632679488005879e-05
7. Cos('a',20)

April 8, 2014

- 4 8. 'Error: parámetro no es número real' ✓  
Cos(1.57075,'3.4')  
'Error: parámetro N debe ser un número entero dentro del intervalo' ✗  
4 9. Cos(0.785,50)  
'Error: parámetro N debe ser un número entero dentro del intervalo' ✓  
4 10. facto(10)  
3628800

- b. La variación de longitud de onda de los fotones dispersados,  $\Delta\lambda$ , puede calcularse a través de la relación de Compton: (10 pts)

$$\Delta\lambda = \frac{h}{m_e c} (1 - \cos\theta),$$

Donde  $h$  es la constante de Planck,  $m_e$  es la masa del electrón,  $c$  es la velocidad de la luz y  $\theta$  el ángulo entre los fotones incidentes y dispersados. La cantidad  $h/m_e c = 0.0243 \text{ \AA}$ , se denomina longitud de onda de Compton.

Construya una función llamada compton(Angulo) que reciba el valor del ángulo  $\theta$  en **grados** y retorne la variación de la longitud de onda de los fotones dispersados, de acuerdo con la relación Compton.

Debe usar la función coseno ya programada por Ud. con  $n=20$ .

**Hint:** Tenga en cuenta que debe convertir el ángulo  $\theta$  de **grados a radianes** para así poder invocar la función  $\cos(X;N)$ . Recuerde que  $180^\circ$  grados =  $3,14159265$  rad.

1. compton(0)  
0 ✗  
2. compton(180)  
0.04859996918097921 ✗  
3. compton(270)  
0.024358052168128502 ✗  
4. compton('90')  
'Error: solo se admiten grados' ✗  
5. compton([])  
'Error: solo se admiten grados' ✗

**Nota técnica:** La diferencia de longitud de onda se obtiene del análisis de la interacción, tratándola como si ésta fuese una colisión elástica, su deducción requiere solamente del uso de los principios de conservación de energía y momentum.

**Nota:** Los programas deben nombrarse de la forma establecida con los argumentos indicados. La calificación se hace con base en el resultado de invocar el nombre de la función con los argumentos y debe responder a los ejemplos establecidos. Se tiene hasta el día Miércoles 9 de Abril 2014 media noche para enviar la versión digital sin cambios de lógica al asistente (únicamente sangría y léxicos cada uno de estos cambios se indica mediante un comentario en el código del programa.