Tarea 1

Métodos Numéricos y Algoritmos Computacionales
Carlos Espinosa

26 de julio de 2021

Instrucciones. Deben realizar, máximo seis problemas de la sección *problemas generales*, máximo dos problemas de la sección de *graficación* y, de la sección *Find-root* realizar los problemas correspondientes para resolver 10 problemas en total.

Problemas generales

Pregunta 1. Un carpintero está acomodando sus tablas que tiene en su taller. Por alguna extraña razón, quiere arreglarlas de la más pequeña a la más grande tal que cada tabla sea mas grande que la anterior por exactamente 1 unidad (arbitraria). Es posible que para cumplir su cometido, el carpintero necesite agregar tablas adicionales. Haz una función que calcule el número mínimo de tablas que el necesitará.

Ejemplo

Para un arreglo de tablas *tablas = [7, 3, 4, 9]*, el resultado de la función debería de ser 3 ya que el necesitaría tablas de tamaño 5, 6 y 8.

Reglas

- No pueden usar paquetes
- Los elementos del arreglo de entrada solo serán números enteros positivos.
- El arreglo de entrada puede contener de 1 a 10 elementos
- Un elemento del arreglo de entrada puede medir de 0 a 20 unidades.
- La salida de la función solo debe de ser un número de entero.

Pregunta 2. Dadas dos *strings*, hacer una función que encuentre el número de caractéres en común.

Ejemplo

Sea *s1 = 'ccell'* y *s2 = czlcc*, la salida de la función debería de ser 3 ya que hay dos c y una l en común.

Reglas

- No pueden usar paquetes
- Las *strings* de entrada solo contendran letras en minúsculas sin incluir la $\tilde{n}$.
- Las *strings* de entrada solo podrán tener de 1 a 15 caractéres.
- La salida solo será un entero.

Pregunta 3. Cierto dueño superticioso de una cafetería aborda todos los días el transporte público para ir a su negocio (todo esto antes de la pandemia). Cada vez que aborda cierto transporte público, el hombre recibe un boleto con un número de serie. El considera que tendrá un buen día si la suma de la primera mitad de los dígitos del número de serie es igual a la suma de los dígitos de la segunda mitad. Dado un número de serie n, hacer una función que le diga si tendrá un buen día o no.

Ejemplo

Si el número de serie es *n=2341*, entonces la función debería de regresar *True*. Para el número de serie *n=248028* la función debería de regresar *False*.

Reglas

- No pueden usar paquetes
- El entero que se dará a la función será un número positivo con un número par de digitos.
- El entero de entrada podrá tomar un valor mínimo de 10 y un valor máximo de 10^6.
- La salida del programa debe de ser un booleano: *True* o *False*.

Pregunta 4. Una señora gusta de coleccionar tarros de galletas. Al organizar sus tarros por tamaño en su vitrina de exhibición, se da cuenta que hay algunos lugares que no puede ocupar dado que la madera de esos entrepaños está en mal estado y sus tarros podrían caerse (dado que ha gastado mucho dinero en los tarros, no puede comprar un nuevo mueble). Haz una función que ordene los tarros sin que ocupen los entrepaños en mal estado.

Ejemplo

Para un arreglo de entrada *tarros = [-1, 15, 19, 17, -1, -1, 16, 18]*, la salida del programa debería de ser *tarros = [-1, 15, 16, 17, -1, -1, 18, 19]*.

Reglas

- No pueden usar paquetes
- Los entrepaños dañados son representados con un -1, por lo que esos elementos no se pueden mover de posición en el arreglo de entrada. Todos los demás números serán positivos y representarán la altura de los tarros de galletas en una unidad arbitraria.
- Los tarros pueden medir de 0 a 1000. Un cero representa que ahí no pondrá ningún tarro
- El arreglo de entrada puede contener de 1 a 1000 elementos
- La salida debe de ser un arreglo con los -1 *intactos* y los tarros acomodados por altura, del más chico al más grande.

Pregunta 5. Dada una string, hacer una función que escriba al revés las palabras que estén dentro de paréntesis. Puede haber paréntesis dentro de paréntesis.

Ejemplos

- Para *'(cap)'*, la salida de la función debe de ser *'pac'*
- Para *'lol(cap)rap'*, la salida de la función debe de ser *'lolpacrap'*
- Para *'lol(cap)rap(tam)'*, la salida de la función debe de ser *'lolpacrapmat'*
- Para *'lol(cap(rap))tam'*, la salida de la función debe de ser *'lolrappactam'*. Esto se debe a que *'lol(cap(rap))tam'* se transforma a *'lol(cappar)tam'* y esto se transforma en *'lolrappactam'*.

Reglas

- No pueden usar paquetes
- La *string* de entrada solo tendrá letras en minúsculas sin incluir la *ñ*.
- Todos los paréntesis tendrán su pareja correspondiente.
- La *string* de entrada podrá estar vacía. Por el contrario, podrá contener hasta 50 caractéres.
- La salida de la función debe de ser la string transformada.

Pregunta 6. Dada una *string* que contiene espacios, hacer un función que sustituya los espacios por guiones medios.

Ejemplo

Dada la *string s1 = 'esto es una string'*, la salida de la función debe de ser *'esto-es-una-string'*.

Reglas

- No pueden usar paquetes
- La *string* de entrada contendrá espacios y podrá contener de 1 a 10 palabras separadas por espacios.
- La salida de la función debe de ser la string transformada.

Pregunta 7. Dada una *string*, hacer una función que nos indíque si contiene caractéres alfanuméricos, caractéres alfabéticos, dígitos, minúsculas y mayúsculas.

Ejemplo

Para la *string s1='qA2'*, la salida debe de ser:
True
True
True
True
True

Reglas

- No pueden usar paquetes
- La *string* de entrada puede contener cualquier símbolo alfanumérico así como símbolos especiales.
- La *string* de entrada puede ser vacía.
- La *string* de entrada puede contener hasta 1000 caractéres
- La función no debe de regresar nada, pero si debe de imprimir *True* o *False* de acuerdo a lo siguiente:
  - La primera línea debe indicar si en la *string* de entrada hay caractéres alfanuméricos
  - La segunda línea debe indicar si en la *string* de entrada hay caractéres alfabéticos.
  - La tercera línea debe indicar si en la *string* de entrada hay digitos numéricos.
  - La cuarta línea debe indicar si en la *string* de entrada hay caractéres en minúscula.
  - La quinta línea debe indicar si en la *string* de entrada hay caractéres en miyúscula.

Pregunta 8. Cierta alumna del curso quiere recompersarse asi misma después de acabar la tarea 2 de MNyAC ordenando algo para cenar. Abré la única[1]aplicación de *delivery* (ya que estamos en pandemia) y empieza a escoger lo que será su cena. En su restaurante favorito hay solamente siete platillos, cuyo precios son 7, 27, 41, 49, 63, 78 y 108 en una moneda arbitraria.

Por suerte, nuestra querida alumna tiene tres cupones:

- Cuando el precio del pedido alcance un total de 69, ella recibirá un descuento de 15.
- Cuando el precio del pedido alcance un total de 89, ella recibirá un descuento de 30.
- Cuando el precio del pedido alcance un total de 120, ella recibirá un descuento de 50.

Pero, ella solo puede usar un cupón a la vez y cada cupón solo puede ser usado una vez. Desgraciadamente, la app siempre escoge el mejor cupón dependiendo de la orden. Por ejemplo, si el precio total es de 300, automáticamente tendrá que pagar solamente 250. Si ella ordena n platillos, tal que P_1, P_2, P_3, $\cdots$, p_n, ella quiere saber cuanto gastará en su cena.

Hagamos una función que ayude a nuestra compañera a decidir que comida pedir. La alumna desea probar varias combinaciones antes de decidir, además ella desea hacer su programa interactivo

[1]Solo hay una ya que hubo una guerra de apps de *delivery* y al final todas se unificaron.

por lo que el único parámetro que debe de recibir la función será el número C de combinaciones que ella quiere probar. Para cada caso a aprueba, la función debe de pedir dos cosas (en líneas diferentes): i) El número n de platillos, y ii) que platillos (P_1, P_2, P_3, $\cdots$, p_n) desea.

La función no debe de regresar nada, pero si debe de imprimir, para cada caso, cuanto dinero gastaría la alumna.

Ejemplo

Si la función recibe $C = 5$, la entrada podría verse algo así:
2
2 3
3
2 3 4
6
1 1 4 5 1 4
1
1
7
7 7 7 7 7 7 7
Y la salida sería:
68
87
132
7
706

En este caso, se debe de usar la función *input()* y, como casi todos los ejercicios de esta sección, no pueden usar paquetes.

Problemas de graficación

Instrucciones. En esta sección, todas las gráficas deben de llevar su correspondientes etiquetas en ambos ejes y un título.

Pregunta 9. Seguir las siguientes instrucciones para generar las gráficas que se piden. En todos los puntos se debe de generar la gráfica que se pide. Por la naturaleza de las preguntas, pueden usar paquetes.

1. Generar tres conjuntos de mil números aleatorios que sigan una distribución normal. Cada conjunto debe estar centrado en diferentes valores (μ_1, μ_2, μ_3) y con diferentes anchuras (σ_1, σ_2, σ_3). Cada conjunto debe de ser guardado en una variable diferente.

2. Hacer un histograma donde se muestren las tres distribuciones correspondientes a los tres conjuntos de números aleatorios. Las distribuciones deben de estar acomodadas de acuerdo a su anchura.

3. Repetir el histograma del punto anterior y añadir líneas verticales que represente el valor correspondiente μ_n en la cual se encuentran centradas las distribuciones.

4. Hacer una gráfica de *scatter* donde se muestren las tres diferentes distribuciones con un color diferente. Agregar una leyenda donde diga que color corresponde a que distribución.

5. Hacer un una gráfica que incluya todas las gráficas anteriores en una misma figura. Cada *subplot* debe de tener su propio título, también la figura principal debe de tener su título correspondiente.

Pregunta 10. Graficar las funciones de los problemas 16 y 17. Para el polinomio, ajustar la gráfica para que se vean las soluciones reales. Para la función trigonométrica, ajustar la gráfica para que se vea la solución dada en el problema 17.

Pregunta 11. Sabemos que el sistema oscilante mas simple que conocemos es una masa y un resorte. Aplicando la segunda ley de Newton $F = ma$ a la masa, podemos obtener las ecuaciones de movimiento del sistema:

$$m\frac{d^2x}{dt^2} + kx = 0$$

que puede ser reescrita como:

$$\frac{d^2}{dt^2} + \omega_0^2 x = 0$$

donde $\omega_0 = \sqrt{k/m}$ es la frecuencia natural de oscilación. Las soluciones a esta ecuación de movimiento es:

$$x(t) = x_m \cos(\omega_0 t + \phi)$$

Donde x_m es la amplitud de oscilación y ϕ es la constance de fase de la oscilación.

1. Hacer una imagen que muestre la pósición de tres diferentes diferentes objetos que sigan la ecuación antes descrita con frecuencias naturales de ω_0, $2\omega_0$ y $3\omega_0$. Suponer que todos los objetos tienen las misma amplitud de oscilación y la misma constante de fase.

2. Suponer que tenemos un objeto que oscila de acuerdo a la ecuación de movimiento antes planteada. Con una frecuencia natural de ω_0 y una constante de fase ϕ. Suponiendo que el sistema inicialmente está en reposo pero desplazado una distance x_m del equilibrio, realizar una figuar con tres *subplots* donde se muestre:

   - La energía cinética
   - La energía potencial
   - La energía total

   Pueden utilizar cualquier paquete que necesitan, exceptuando aquellos que generen automáticamente las gráficas requeridas.

Problemas *Find-root*

Pregunta 12. Encontrar las soluciones de las siguientes ecuaciones con uno de los tres métodos vistos en clase (indicar con que método están encontrando las soluciones). **Pueden usar paquetes**.

1. $x - 2x - 5 = 0$ en $[1, 4]$
2. $x - \cos x = 0$ en $[0, \pi/2]$
3. $x + 3x - 1 = 0$ en $[-3, -2]$
4. $x - 0.8 - 0.2 \sin x = 0$ en $[0, \pi/2]$

Pregunta 13. Encontrar las soluciones con una precisión de al menos 10^{-5} para las siguientes ecuaciones con cualquier método de los visto en clase. **No pueden usar paquetes que resuelvan las ecuaciones, pueden usar el paquete *numpy* pero no las rutinas de este para encontrar las soluciones**.

1. $2x \cos 2x - (x - 2) = 0$ en $[2, 3]$ y en $[3, 4]$
2. $(x - 2) - \ln x = 0$ en $[1, 2]$ y en $[e, 4]$
3. $e^x - 3x = 0$ en $[0, 1]$ y en $[3, 5]$
4. $\sin x - e^{-x} = 0$ en $[0, 1]$ y en $[6, 7]$

Pregunta 14. Encontrar las cuatro soluciones de $4x \cos(2x) - (x - 2) = 0$ en $[0, 8]$ con una precisión de al menos 10^{-5}. Pueden usar paquetes siempre y cuando garantizen la precisión de alguna forma. Usar uno de los métodos vistos en clase.

Pregunta 15. Encontrar todas las soluciones de $x + \cos x = 0$ con una precision de al menos 10^{-5}. Pueden usar paquetes siempre y cuando garantizen la precisión de alguna forma. Usar uno de los métodos vistos en clase.

Pregunta 16. El polinomio de cuarto grado

$$f(x) = 230x^4 + 18x^3 + 9x^2 - 221x - 9$$

tiene dos soluciones reales, uno en $[-1, 0]$ y otro en $[0, 1]$. Calcular una aproximación de estas soluciones con una presición de 10^{-6} con dos de los tres métodos vistos en clase. Pueden usar paquetes siempre y cuando aseguren que están usando los métodos indicados.

Pregunta 17. La función $f(x) = \tan \pi x - 6$ tiene una solución en $(1/\pi) \arctan 6 \approx 0.447431543$. ¿Cuál de los tres métodos vistos en clase es mejor para encontrar la solución? ¿Por qué?. Pueden usar paquetes siempre y cuando aseguren que están usando los métodos indicados.