

Midterm – 1 Sem 2014

Pregunta 1

Escribe una función llamada *trioPitagorico* que reciba como parámetro un número entero positivo a y determine si existen otros dos números enteros positivos, b y c , tales que:

$$a > b > c$$

es decir, b y c son menores que a y además son distintos entre ellos, y

$$a^2 = b^2 + c^2$$

es decir, a , b y c forman un *trío pitagórico*: si a fuera la longitud de la hipotenusa de un triángulo rectángulo, entonces b y c serían las longitudes de los catetos correspondientes

La función debe devolver una lista con los dos valores, b y c , o bien una lista vacía, si b y c no existen para el a dado.

Ejemplo:

trioPitagorico(5) debe devolver la lista **[4, 3]**, ya que $5^2 = 4^2 + 3^2$; pero *trioPitagorico*(6) debe devolver la lista vacía, ya que no hay dos números enteros positivos menores que 6 tales que la suma de sus cuadrados sea igual a $6^2 = 36$.

Pregunta 3

Twitter es una red social que permite enviar mensajes de hasta 140 caracteres de largo. Para facilitar la clasificación de los mensajes es posible incluir hashtags, que son palabras anteceditas por un #. Con ellos el sistema genera una lista de Trending Topics, que son los hashtags más usados.

Se te pide programar un mini Twitter, con las siguientes funcionalidades:

contieneHashtag(mensaje): Recibe un mensaje y retorna **True** si contiene uno o más hashtags y **False** en otro caso. Por ejemplo, *contiene_hashtag("Me encanta #programar")* retorna True, y *contiene_hashtag("Me encanta programar")* debe retornar **False**.

obtenerHashtags(mensaje): Recibe un mensaje, y retorna todos los hashtags que tiene ese mensaje como un string separado con un espacio entre cada uno. Ejemplo: si recibe "Me encanta el #verano y #programar" debe retornar *verano programar*.

ponerHashtag(mensaje, hashtags): Recibe un string que representa el mensaje a publicar y un string compuesto por hashtags separados por comas (','), Debe retornar el mensaje con # insertados antes de las palabras recibidas en hashtags. Ejemplo, dado el mensaje "Me encanta el verano y programar" y *verano,programar*, debe retornar "Me encanta el #verano y #programar"

actualizarTrendingTopics(hashtag): Nuestro Twitter está interesado en contar solo los hashtags "verano", "programar" e "IIC1103". Esta función recibe una palabra y, en caso de ser alguna de las palabras de interés, actualiza la cantidad de menciones de ese hashtag. Es decir, si se llama a *actualizar_trending_topics("programar")*, debe aumentar en 1 la cantidad de menciones de la palabra "programar". Declara las variables adecuadas para llevar esta cuenta.

twittear(mensaje, truncar): Si *mensaje* contiene las palabras "verano", "programar" y/o "IIC1103" (sin hashtags), debes usar las funciones definidas anteriormente para agregar los hashtags al mensaje. Una vez hecho esto, o bien si el mensaje original ya tenía los hashtags, debes actualizar las menciones de cada trending topic y mostrar un mensaje con la cantidad de veces que se ha mencionado cada una de esas palabras. Si el mensaje no contiene las palabras de interés, no se debe mostrar el conteo de menciones.

Si el mensaje procesado como se señala anteriormente excede los 140 caracteres:

- Si *truncar* es **True**, retorna el mensaje hasta el caracter 140.
- Si *truncar* es **False**, retorna los caracteres sobrantes, desde el 141 en adelante

En caso de que el mensaje no exceda los 140 caracteres, se retorna el mensaje completo.

Puedes asumir que cada palabra de interés aparecerá a lo más una sola vez por mensaje. Recuerda, además, que las palabras de interés pueden venir en mayúscula o minúscula

Pregunta 4

La administración de información académica necesita un programa que permita obtener los promedios y las estadísticas de cursos de la universidad. Por este motivo se te pide que hagas lo siguiente:

- a. Implementa la función de nombre *calcularNotaPromedio* que reciba como parámetro las notas que un alumno obtiene en un curso, y que calcule y retorne el promedio ponderado de estas notas. Para esto las notas de un alumno están definidas como una lista de la forma: [(nota11,ponderacion11), (nota12,ponderacion12), ...]. Considera que cuando un alumno no tiene todas sus notas el promedio no se puede calcular y por ende debes dejar la nota en categoría de *nota pendiente* (np).

Por ejemplo, si un alumno tiene las siguientes notas en el curso [(7,0.3), (5,0.4), (4.8,0.3)] entonces indica que alumno tiene 3 notas, un 7 con ponderación 30%, un 5 con ponderación de un 40% y un 4.8 con ponderación de un 30%. Para este caso, el resultado que se obtiene al invocar la función *calcularNotaPromedio* es 5.5