

Práctica 1 de Programación 1

Daniella Zambrano y Jesús Pérez

Noviembre, 2019

Fechas válidas de entrega: desde el 17/11/2019 hasta el 24/11/2019. Su entrega consiste de un archivo llamado *horas.c*. Por favor, colocar en comentario al inicio de este archivo, su número de cédula, nombre y apellido.

1 Introducción

El problema a resolver consiste en un convertidor de Tiempo. El cual usted, debe programar y usar un conjunto de funciones que permitan resolver pequeños problemas para llevar a cabo las transformaciones de tiempo.

Para tener el convertidor de Tiempo se requiere de un tiempo inicial que podrá ser expresado en un formato “**hh:mm:ss**” o en una cantidad de segundos “**ss**”. El objetivo será hacer las respectivas transformaciones para poder generar el formato contrario al utilizado para la entrada.

Un ejemplo de esto sería:

-Dado el formato 02:38:12, donde 02 es hh, 38 es mm y 12 es ss, la salida se deberá expresar en el formato “ss”, es decir 9492.

-Dado el formato 64830, la salida se deberá expresar en el formato “hh:mm:ss”, es decir, 18:00:30.

2 Práctica

Para la realización de este trabajo, se deberá hacer uso de lo visto en clases, sin añadir ninguna otra biblioteca adicional a “pr1-ula.h” y sin añadir contenido adicional a lo explicado en el curso hasta los momentos.

Casos de prueba:

Input: 14:00:49

Output: 50449

Input: 07:09:00
Input: 70:15:36
Input: 15831
Input: 72000

Output: 25740
Output: 252936
Output: 04:23:51
Output: 20:00:00

Estos casos de prueba le servirán como una ayuda para verificar que su programa funcione bien, pero los evaluados serán distintos a los aquí expresados.

3 Enunciado

La empresa CONFIANZA requiere la creación de un programa que convierta formato de hora a formato en segundos, ya que dicha empresa tiene un robot que por cuestiones de límite de 4 bytes de espacio de memoria solo puede almacenar un valor entero y por lo tanto, representa el tiempo en un contador de segundos. A pesar de esto sus clientes requieren poder visualizar el tiempo transcurrido en un formato hh:mm:ss, para ello se le proveen las siguientes rutinas.

4 Rutinas

1. Calentamiento:

a) **void imprimirHora(Entero)** la cual recibe un entero hora y solamente deberá imprimir su valor por terminal. Esta rutina se deberá realizar en una sola instrucción de código.

b) **void imprimirMin(Entero)** la cual recibe un entero minuto y solamente deberá imprimir su valor por terminal. Esta rutina se deberá realizar en una sola instrucción de código.

c) **void imprimirSec(Entero)** la cual recibe un entero segundo y solamente deberá imprimir su valor por terminal. Esta rutina se deberá realizar en una sola instrucción de código.

2. Programa:

a) **void convertirASec(Entero,Entero,Entero)** la cual recibe como parámetros un entero hora, un entero minuto y un entero segundo, y éstos deberán ser convertidos a un tiempo expresado en solo segundos. La realización de esta rutina no debe tener ningún `imprimirEntero(Entero)`, en este caso, debe hacer uso de la rutina anteriormente programada (`imprimirSec`).

b) **void convertirAFormat(Entero)** la cual recibe como parámetro un entero segundo,

el cual deberá ser transformado al formato hh:mm:ss. Se debe tener en cuenta que, la cantidad en ss no puede exceder de 59, ya que 60 sería la adición en 1 de mm. El mm no debe exceder de 59, ya que el 60 representa una hora más. La realización de esta rutina no debe tener ningún imprimirEntero(Entero), en este caso, debe hacer uso de las rutinas anteriormente programadas.

c) **void calHoras(Entero, Entero)** la cual recibe dos enteros que expresan el tiempo en segundos, su deber será programar la solución para encontrar la diferencia de tiempo entre estos dos, recordando que diferencias de tiempo no pueden ser negativas, la respuesta deberá ir expresada en el formato hh:mm:ss. La realización de esta rutina no debe tener ningún imprimirEntero(Entero), en este caso, debe hacer uso de las rutinas anteriormente programadas.

5 Evaluación

La fecha de entrega de esta práctica es desde el 17/11/2019 hasta el 24/11/2019.

Para la evaluación se debe enviar el archivo *horas.c* a la dirección:

daniellamzr@gmail.com

El “asunto del correo” debe ser exactamente el texto **“PR1-LAB-01”** sin las comillas. Si falla con el asunto del correo entonces su práctica no será evaluada, ya que el mecanismo automatizado de filtrado no podrá detectar su trabajo. NO comprima el archivo y no envíe nada adicional a éste. El único contenido que debe aparecer en el correo es el archivo “horas.c” en archivo adjunto, y en el mensaje, su sección, el primer apellido, el primer nombre, como se muestra en el siguiente ejemplo:

S1,Perez,Jesus

Durante el periodo de entrega usted tiene dos intentos para la corrección de su práctica, de modo que, si en el primer intento no obtuvo la calificación máxima, se le responderá el correo con los fallos para que los acomode y la vuelva a enviar, y el segundo intento será el que tendrá la calificación final de la práctica. **Nota:** Tenga presente el rango de entrega para sus dos intentos, considerando que en el primer intento el envío de los fallos no será inmediato.

Por favor, no use otros medios para enviar la solución ni escriba otros comentarios adicionales en el email. **Atención:** si su programa no compila, entonces el evaluador no compila. Si una de las rutinas cae en un bucle infinito entonces el evaluador cae en un bucle infinito. Por esa razón, en todos estos casos será imposible darle una nota, lo que simplemente se traduce en que usted tiene cero en el intento en el cual ocurra una de las circunstancias mencionadas. No envíe programas que no compilan.

6 Recomendaciones

1. Lea enteramente este enunciado antes de proceder al diseño e implementación. Asegúrese de comprender bien su diseño. Haga un boceto de su programa y como es esperado que funcione. Por cada rutina, plantee qué es lo que va a hacer, cómo va a resolver el problema. Esta es una situación que amerita de diseño y desarrollo.

2. Use el grupo de Telegram para plantear las dudas de comprensión. Pero de ninguna manera comparta código, pues es considerado **plagio**.

3. No se debe modificar la plantilla que se envía ni tampoco agregar parámetros adicionales, pero al añadir nuevas variables asignele nombres que ayuden a comprender el funcionamiento del código.