

Actividad #1

Calificación: N/A



Vence: Saturday, 25 November, 2017 at 8:00 am

Entregas

Enviar Tarea

Escribir 11 funciones SML relacionadas a fechas calendario. En todos los problemas, una "fecha" es un valor SML de tipo `i int*int*int`, donde la primera parte es el año, la segunda parte el mes y la tercera parte es el día.

Una fecha "razonable" tiene un año positivo, un mes entre 1 y 12 y un día no mayor a 31 (o menos dependiendo del mes). Un "día del año" es un número entre 1 y 365 donde por ejemplo, 33 representa el 2 de febrero.

1. Escribir una función **is_older** que tome dos fechas y las evalúa a true o false. Evalúa a true si el primer argumento es una fecha que viene antes del segundo argumento. (Si las dos fechas son iguales, el resultado es falso).
2. Escribir una función **number_in_month** que toma una lista de fechas y un mes (i.e., un int) y retorna cuantas fechas en la misma lista hay en ese mes.
3. Escribir una función **number_in_months** que toma una lista de fechas y una lista de meses (i.e., una int list) y retorna el número de fechas en la lista de fechas que están en cualquiera de los meses en la lista de meses. Asumir que la lista de meses no tiene números repetidos. (Nota: utiliza la respuesta de la pregunta anterior).
4. Escribir una función **dates_in_month** que toma una lista de fechas y un mes (i.e., un int) y retorna una lista que contiene las fechas del argumento de lista de fechas que hay en el mes. La lista retornada debe contener las fechas en el orden que se le han dado originalmente.
5. Escribir una función **dates_in_months** que toma una lista de fechas y una lista de meses (i.e., an int list) y retorna una lista que contenga las fechas del argumento lista de fechas que están en cualquiera de los meses en la lista de meses. Asumir que la lista de meses no tiene números repetidos. Nota: utilizar la respuesta del problema anterior y el operador "@".
6. Escribir una función **get_nth** que tome una lista de strings y un int n y retorne el enésimo elemento de la lista donde la cabeza de la lista es el primero. Nota: la función puede aplicar hd o tl a la lista vacía.
7. Escribir una función **date_to_string** que toma una fecha y retorna un string de la forma January 20, 2013. Utilizar el operador para concatenar strings y la función Int.toString de la librería para convertir un int a string. Para producir el mes, utilizar una lista con 12 strings y la respuesta al problema anterior.
8. Escribir una función **number_before_reaching_sum** que tome un int llamado sum, el cual se puede asumir que es positivo, y un int list, el cual se puede asumir que contiene todos sus números positivos y retorna un int. Se debería retornar un int n tal que los primeros n elementos de la lista sumen menos que la suma, pero el primer n+1 elemento de la lista suma a suma. Asumir que toda la lista suma más que el valor entregado en el valor; está bien si ocurre una excepción si este no es el caso.
9. Escribir una función **what_month** que toma un día del año (i.e., un int entre 1 y 365) y retorna en que mes se encuentra ese día (1 para enero, 2 para febrero, etc.). Utilizar una lista que tenga 12 integers y la respuesta al anterior problema.
10. Escribir una función **month_range** que toma dos días del año day1 y day2 y retorna una int list [m1,m2,...,mn] donde m1 es el mes del day1, m2 es el mes del day1+1, ..., y mn es el mes del día day2. Nota: el resultado tendrá la longitud day2 - day1 + 1 o longitud 0 si day1 > day2.
11. Escribir una función **oldest** que toma una lista de fechas y evalúa a una opción (int*int*int). Evalúa a NONE si la lista no tiene fechas y SOME d si la fecha es la más antigua de la lista.

Hoy a 12:04 pm publicado