Instituto Tecnológico de Costa Rica
Lenguajes de Programación
Tarea 3

Calentando motores con SML/NJ

Se van a escribir algunas funciones SML relacionadas a fechas. En todos los casos la fecha va a ser de tipo int*int*int, donde el primer entero será el año, el segundo entero el mes y el último entero el día. El año será un número positivo, el mes un número entre 1 y 12, y el día entre 1 y 31 (o menos dependiendo del mes). Un "día del año" es un número entre 1 y 365, donde, 33 representaría el 2 de febrero.

Problemas:

1. Escriba una función *is_older* que toma dos fechas y retorna un booleano. El retorno es true si el primer argumento es una fecha que está antes de la fecha del segundo argumento. (Si ambas fechas son iguales, el retorno será false).

2. Escriba una función *number_in_month* que toma una lista de fechas y un mes (el mes sería un int) y retorna un número que indica cuántas fechas están en el mes dado.

3. Escriba una función *number_in_months* que toma una lista de fechas y una lista de meses (una lista de enteros) y retorna un int que indica el número de fechas que están en cualquiera de los meses en la lista de meses. (Nota: use la función del problema anterior).

4. Escriba una función *dates_in_months* que toma una lista de fechas y un mes (un entero) y retorna una lista de fechas, que serían las fechas en las que ese mes está presente. La lista que se retorna debe contener las fechas en el mismo orden en que van apareciendo en la lista original.

5. Escriba una función *dates_in_months* que toma una lista de fechas y una lista de meses (una lista de enteros) y retorna una lista de fechas que están en cualquiera de los meses de la lista de meses. (Nota: use la función del problema anterior. También investigue y use el operador @).

6. Escriba una función *get_nth* que toma una lista de strings y un entero ***n*** y retorna el i-ésimo elemento de la lista, donde la cabeza de la lista es el primer elementos (i = 1). No se preocupe por el caso en el que la lista tiene pocos elementos: su función puede aplicar el hd o tl a la lista vacía en ese caso.

7. Escriba una función *date_to_string* que toma una fecha y retorna un string de la forma **Enero 20, 2015** (por ejemplo). Use el operador **^** para concatenar strings y la función **Int.toString** de la biblioteca de SML para convertir un **int** a **string**. Para producir la parte del mes, no use un montón de condicionales; en su lugar, use una lista con 12 strings y la función que se utilizó en el problema anterior. Por consistencia, ponga una coma después del día y los meses con su inicial mayúscula: Enero, Febrero, etc.

8. Escriba una función *number_before_reaching_sum* que toma un **int** llamado **sum**, el cual es positivo, y un **int list**, el cual se asume son enteros positivos; y al final retorna un **int**. Se debe retornar un **int n**, tal que los primeros **n** elementos de la lista suman menos que **sum**, pero los primeros **n+1** elementos de la lista suman **sum** o más.

9. Escriba una función *what_month* que toma un dia del año (un entero entre 1 y 365) y retorna a cuál mes ese día pertence (1 para Enero, 2 para Febrero, etc.). Use una lista conteniendo 12 enteros y la función del problema anterior.

10. Escriba una función *month_range* que toma dos días del año día1 y día2 y retorna una lista [m1, m2, …, mn] donde m1 es el mes de día1, m2 es el mes de día1 + 1, …, mn es el mes del día2. Note que los resultados van a tener una longitud de día2 – día1 + 1 o longitud 0 si día1 > día2.

11. Escriba una función *oldest* que toma una lista de fechas y evalúa a una option (int*int*int). Evalúa a NONE si la lista no tiene fechas y a SOME d si la fecha d es la más vieja en la lista.

A continuación se listas los casos de prueba:

- val test1 = is_older ((1,2,3),(2,3,4)) = true
- val test2 = number_in_month ([(2012,2,28),(2013,12,1)],2) = 1
- val test3 = number_in_months ([(2012,2,28),(2013,12,1),(2011,3,31),(2011,4,28)],[2,3,4]) = 3
- val test4 = dates_in_month ([(2012,2,28),(2013,12,1)],2) = [(2012,2,28)]
- val test5 = dates_in_months ([(2012,2,28),(2013,12,1),(2011,3,31),(2011,4,28)],[2,3,4]) = [(2012,2,28),(2011,3,31),(2011,4,28)]
- val test6 = get_nth (["hi", "there", "how", "are", "you"], 2) = "there"
- val test7 = date_to_string (2013, 6, 1) = "June 1, 2013"
- val test8 = number_before_reaching_sum (10, [1,2,3,4,5]) = 3
- val test9 = what_month 70 = 3
- val test10 = month_range (31, 34) = [1,2,2,2]
- val test11 = oldest([(2012,2,28),(2011,3,31),(2011,4,28)]) = SOME (2011,3,31)

Notas finales

- Esta tarea puede ser resuelta en parejas. Pero hay una condición: ambas personas deben dominar el código, de lo contrario habrá una rebaja de puntos.
- Crear un solo archivo con la solución de la tarea.
- Hacer los casos de prueba.
- Fecha de entrega: viernes 9 de marzo.