



Práctica 4: Práctica Complementaria para el Parcial

1. Escriba definiciones en ML de las siguientes funciones. En cada caso, pueden utilizar funciones auxiliares para facilitar la definición de la función pedida:
 - a. **cubos**: dada una lista de enteros, devuelve una lista con los cubos de cada elemento. *Ejemplo*: `cuadrados([1, 2, 3]) = [1, 8, 27]`.
 - b. **orLista**: dada una lista de booleanos, devuelve `true` si al menos un elemento de la de la lista es `true`. *Ejemplos*: `orLista([true, true, false]) = true` y `orLista([false, false]) = false`.
 - c. **primerosLista**: dada una lista de listas de enteros, devuelva una lista con el primer elemento de cada lista. Si alguna de las listas es vacía, poner un `-1` en ese caso. *Ejemplo*: `primerosLista [[1, 2, 3], [4, 5], [], [6, 7]] = [1, 4, -1, 6]`.
 - d. **longitudPalabras**: dada una lista de strings, devuelve una lista con la longitud de cada string en la lista argumento. *Ejemplo*: `longitudPalabras ["hola", "ser", "cuadro"] = [4, 3, 6]`
 - e. **elim**: dados un entero y una lista de enteros, devuelve la lista luego de eliminar todas las apariciones del entero dado. *Ejemplo*: `elim(4, [1, 2, 3, 4, 5, 6, 4, 7]) = [1, 2, 3, 5, 6, 7]`
2. Queremos definir la función `repetir`, que dado un entero `n`, nos devuelva una lista de longitud `n`, con todos sus elementos iguales a `n`. *Es decir*: `repetir(8) = [8, 8, 8, 8, 8, 8, 8, 8]`, `repetir(3) = [3, 3, 3]`, `repetir(0) = []`.
 - a. Definir `repNM`, que dados un `n` y un `m`, repite `m` veces el número `n` en una lista. *Ejemplo*: `repNM(5, 3) = [5, 5, 5]`.
 - b. Utilizar `repNM` para definir `repetir`.
3. 2. Analice la siguiente función `alts` definida a continuación:

```
fun alts ([], xs, ys) = (xs, ys)
  | alts ([x], xs, ys) = (x::xs, ys)
  | alts (x::y::t, xs, ys) = alts (t, x::xs, y::ys);
```

 - a. Dé el tipo de la función `alts`.
 - b. Explique, paso por paso, la ejecución de la función: `alts([2, 4, 2, 1, 5], [], [])`.
 - c. ¿Qué hace la función `alts`?
4. Defina una función `letraAparece` que dados un carácter y una cadena, defina si el carácter aparece en la cadena o no. Es decir: `letraAparece("#a", "hola") = true` y `letraAparece("#b", "hola") = false`.

Ayuda: puede ser conveniente definir una función auxiliar que tome un carácter y una lista de caracteres como argumentos, y utilizar esa función para definir `letraAparece`.
5. Dado el siguiente tipo de dato que nos permite representar cantidades de dinero en pesos, euros y dólares:

```
- datatype Dinero = Euros of real | Pesos of real | Dolares of real
```

y conociendo las siguientes equivalencias:
 - 1 Peso = 15 Dólares
 - 1 Peso = 17 Euros

Defina las siguientes funciones que tomen una entrada de tipo de dato Dinero y devuelvan un resultado de tipo Dinero según corresponda:

- a. Defina la función `convertir_a_euros` que tome un valor de tipo Dinero y devuelva su equivalente en euros, teniendo en cuenta las equivalencias propuestas más arriba.
- b. Defina la función `convertir_a_pesos` que tome un valor de tipo Dinero y devuelva su equivalente en pesos, teniendo en cuenta las equivalencias propuestas más arriba.
- c. Defina la función `convertir_a_dolares` que tome un valor de tipo Dinero y devuelva su equivalente en dólares, teniendo en cuenta las equivalencias propuestas más arriba.