



Nombre y Apellido:

Mail:

## Prueba

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. **pares**: dada una lista de enteros, devuelve una lista con los elementos pares de la lista. *Ejemplo*: `pares([1, 2, 3, 4, 5, 6]) = [2, 4, 6]`.
  - b. **cortas**: dada una lista de strings, devuelve una lista con aquellas palabras de longitud menor a 5. *Ejemplo*: `cortas(["Lista", "de", "palabras", "sin", "sentido"]) = ["de", "sin"]`.
  - c. **pegar**: dada una lista de strings, devuelve el string que se obtiene de concatenar todos los elementos de la lista. *Ejemplo*: `pegar(["Las ", "lis", "tas ", "son", " complicadas", "."]) = "Las listas son complicadas."`
2. Vamos a trabajar ahora con listas de pares ordenados de reales, los cuales representan puntos del plano:
  - a. Defina la función **distancias** que tome una lista de puntos del plano y devuelva una lista con la distancia al origen de cada uno. *Ejemplo*: `distancias [(3.0, 4.0), (0.0, 2.0), (0.0, 1.0)] = [5.0, 2.0, 1.0]`.
  - b. Defina la función **sumdist** que tome una lista de puntos del plano y devuelva la suma de sus distancias al origen. *Ejemplo*: `sumadist [(3.0, 4.0), (0.0, 2.0), (0.0, 1.0)] = 8.0`.

Ayuda: puede ser conveniente definir una función auxiliar **distancia** que tome un par ordenado que represente un punto del plano y devuelva su distancia al origen de coordenadas. Dado un punto  $(x, y)$  su distancia al origen se calcula de la siguiente manera:  $\sqrt{x^2 + y^2}$ .

3. Analice la siguientes funciones **pertenece** y **dif** definidas a continuación:

```
fun pertenece(x, []) = false
  | pertenece(x, y::t) = if (x=y) then true else pertenece(x, t);
```

```
fun dif([], y) = []
  | dif(x::z, y) = if true = pertenece(x, y) then dif(z, y) else x::dif(z, y);
```

- a. Dé el tipo de las funciones **pertenece** y **dif**.
  - b. Explique, paso por paso, la ejecución de la función: `dif([1, 2, 3], [2, 3, 4])`.
  - c. ¿Qué hace la función **dif**?
4. Dado el siguiente tipo de dato que nos permite representar temperaturas en grados celsius y fahrenheit:  
`- datatype Temperatura = Celsius of real | Fahrenheit of real`  
y sabiendo que para calcular una temperatura en grados Fahrenheit se debe tomar la temperatura en grados Celsius, mutiplicarla por  $(9/5)$  y finalmente sumarle 32.

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

- a. Defina la función `convertir_a_fahrenheit` que tome un valor de tipo Temperatura y devuelva su equivalente en grados fahrenheit.
- b. Defina la función `convertir_a_celsius` que tome un valor de tipo Temperatura y devuelva su equivalente en grados celsius. Tómese su tiempo para reflexionar bien cómo hacer esta conversión.