Funcional 3: Tuplas

Tuplas

Una tupla es un tipo de dato compuesto, que contiene una cantidad fija de elementos y se pueden combinar distintos tipos de elementos, cada elemento está en una posición fija de la estructura.
Ej: Un alumno viene dado por una tupla de 2 elementos, que está formada por el (legajo, promedio).

```
alumno = (“1104433-8”, 9).
```

Existen algunas funciones predefinidas para obtener cada uno de los elementos de la tupla. Por Ej. si quiero obtener el legajo del alumno puedo hacer lo siguiente:

```
Main> fst alumno
"1104433-8"
```

Y si quiero obtener el promedio:

```
Main> snd alumno
9
```

Otros Ejemplo:
fst(4,6) = 4 -- devuelve el primer elemento de la tupla.
snd(5,7) = 7 -- devuelve el segundo elemento de la tupla.

Ejercicios

1. Definir las funciones **`fst3`**, **`snd3`**, **`trd3`**, que dada una tupla de 3 elementos devuelva el elemento correspondiente, p.ej.

   ```
   Main> snd3 (4,5,6)
   5
   Main> trd3(4,5,6)
   6
   ```

2. Definir la función **`aplicar`**, que recibe como argumento una tupla de 2 elementos con funciones y un entero, me devuelve como resultado una tupla con el resultado de aplicar el elemento a cada una de la funciones, ej:

   ```
   Main> aplicar (doble,triple) 8
   (16,24)
   Main> aplicar ((3+),(2*)) 8
   (11,16)
   ```

3. Definir la función **`cuentaBizarra`**, que recibe un par y: si el primer elemento es mayor al segundo devuelve la suma, si el segundo le lleva más de 10 al primero devuelve la resta 2do – 1ro, y si el segundo es más grande que el 1ro pero no llega a llevarle 10, devuelve el producto. Ej:
   Main> cuentaBizarra (5,8)
   40
   Main> cuentaBizarra (8,5)

13
Main> cuentaBizarra (5,29)
24

4. Representamos las notas que se sacó un alumno en dos parciales mediante un par (nota1,nota2), p.ej. un patito en el 1ro y un 7 en el 2do se representan mediante el par (2,7).
   A partir de esto:
   a. Definir la función **`esNotaBochazo`**, recibe un número y devuelve True si no llega a 6, False en caso contrario. No vale usar guardas.
   b. Definir la función **`aprobo`**, recibe un par e indica si una persona que se sacó esas notas aprueba. Usar `esNotaBochazo`.
   c. Definir la función **`promociono`**, que indica si promocionó, para eso tiene las dos notas tienen que sumar al menos 15 y además haberse sacado al menos 7 en cada parcial.
   d. Escribir una consulta que dado un par indica si aprobó el primer parcial, usando `esNotaBochazo` y composición. La consulta tiene que tener esta forma (p.ej. para el par de notas (5,8))
      ```
      Main> (... algo ...) (5,8)
      ```

5. Siguiendo con el dominio del ejercicio anterior, tenemos ahora dos parciales con dos recuperatorios, lo representamos mediante un par de pares ((parc1,parc2),(recup1,recup2)).
   Si una persona no rindió un recuperatorio, entonces ponemos un "-1" en el lugar correspondiente.
   Observamos que con la codificación elegida, siempre la mejor nota es el máximo entre nota del parcial y nota del recuperatorio.
   Considerar que vale recuperar para promocionar. En este ejercicio vale usar las funciones que se definieron para el anterior.
   a. Definir la función **`notasFinales`** que recibe un par de pares y devuelve el par que corresponde a las notas finales del alumno para el 1er y el 2do parcial. P.ej.
      ```
      Main> notasFinales ((2,7),(6,-1))
      (6,7)
      Main> notasFinales ((2,2),(6,2))
      (6,2)
      Main> notasFinales ((8,7),(-1,-1))
      (8,7)
      ```
   b. Escribir la consulta que indica si un alumno cuyas notas son ((2,7),(6,-1)) recursa o no. O sea, la respuesta debe ser True si recursa, y False si no recursa. Usar las funciones definidas en este punto y el anterior, y composición. La consulta debe tener esta forma:
      ```
      Main> (... algo ...) ((2,7),(6,-1))
      ```
   c. Escribir la consulta que indica si un alumno cuyas notas son ((2,7),(6,-1)) recuperó el primer parcial. Usar composición. La consulta debe tener esta forma:
      ```
      Main> (... algo ...) ((2,7),(6,-1))
      ```

   d. Definir la función **`recuperoDeGusto`** que dado el par de pares que representa a un alumno, devuelve True si el alumno, pudiendo promocionar con los parciales (o sea sin recup.), igual rindió al menos un recup. Vale definir funciones auxiliares. Hacer una definición que no use pattern matching, en las eventuales funciones auxiliares tampoco; o sea, manejarse siempre con fst y snd.

6. Definir la función **`esMayorDeEdad`**, que dada una tupla de 2 elementos (persona, edad) me devuelva True si es mayor de 21 años y False en caso contrario. Por Ej:.

```
Main> esMayorDeEdad (juan,18)
False
```

   Nota: Definir la función utilizando aplicación parcial y composición.

7. Definir la función calcular, que recibe una tupla de 2 elementos, si el primer elemento es par lo duplica, sino lo deja como está y con el segundo elemento en caso de ser impar le suma 1 y si no deja esté último como esta.

```
Main> calcular (4,5)
(8,6)
Main> calcular (3,7)
(3,8)
```

   Nota: Resolverlo utilizando aplicación parcial y composición.