Tema 2 – Listado II de ejercicios

1. Implementar una función que dada una cadena de caracteres y un carácter, indique el número de apariciones del carácter en la cadena. Se debe utilizar ajuste de patrones. Pista: utilizar la definición de listas por comprensión. Ejemplos de aplicación de la función podrían ser los siguientes:

```
> contarApariciones "casa" 'c'
1
> contarApariciones "casa" 'a'
2
> contarApariciones "" 'c'
0
```

2. Implementar una función que dada una 3-tupla, donde cada uno de ellos es a su vez una tupla de dos elementos de tipo `String` e `Int`, respectivamente, retorne una 3-tupla que contenga el primer elemento de cada tupla interna. Se deberá utilizar ajuste de patrones. Un ejemplo de aplicación de la función podría ser el siguiente:

```
> manipula3Tuplas (("hola", 1), ("hello", 2), ("ciao", 3))
("hola","hello","ciao")
```

3. Implementar una función que devuelva `True` si la suma de los cuatro primeros elementos de una lista de enteros es un valor menor a 10 y devuelva `False` en caso contrario. Se deberá utilizar ajuste de patrones. Ejemplos de aplicación de la función podrían ser los siguientes:

```
> sumaMenor10 [1,1,3,4,6,7,8]
True
> sumaMenor10 [1,4,5,2,7,8]
False
```

4. Implementar una función que dado un carácter, que representa un punto cardinal, devuelva su descripción. Por ejemplo, dado 'N' devuelva "Norte". Ejemplos de aplicación de la función podrían ser

```
> puntoCardinal 'N'
"Norte"
> puntoCardinal 'C'
"El caracter introducido no pertenece a un punto cardinal"
```

5. Implementar una función que dada una frase retorne un mensaje donde se indique cuál es la primera y la última letra de la frase original. Un ejemplo de aplicación de la función podría ser

```
> mensajeFrase "Ejercicios del Tema 3"
"La primera letra de la frase Ejercicios del Tema 3 es 'E' y la ultima
letra es '3'"
```

6. Implementar una función que dado un número entero n y una lista de enteros, devuelva True si todos los elementos de la lista son iguales a n. No puedes utilizar recursividad. Ejemplos de aplicación de la función podrían ser:

```
> todosIguales 7 [7, 7, 7]
True
> todosIguales 7 [1, 7, 2, 6]
False
> todosIguales 7 []
False
```

7. Implementar una función que dado un número entero devuelva mensajes indicando en qué rango de valores se encuentra dicho número (menor de 10, entre 10 y 20 o mayor de 20). Se deben utilizar definiciones locales. Ejemplos de aplicación de la función podrían ser

```
> clasificarValorEntrada 20
"El valor de entrada es mayor o igual a 10 y menor o igual a 20"
> clasificarValorEntrada 9
"El valor de entrada es menor que 10"
> clasificarValorEntrada 35
"El valor de entrada es mayor que 20"
```

8. Se considera que dos números son amigos cuando la suma de los divisores propios (todos los divisores salvo el propio número) de cada uno de ellos da como resultado el otro número. Por ejemplo, los números 220 y 284 son números amigos ya que se cumple lo siguiente:

   - La suma de los divisores propios de 220 es 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284
   - La suma de los divisores propios de 284 es 1 + 2 + 4 + 71 + 142 = 220

   Se pide, implementar una función en Haskell que devuelva True si los números de una 2-tupla son amigos. Ejemplos de aplicación de la función podrían ser:

```
> amigos (15, 25)
False
> amigos (220, 284)
True
```

9. Implementar una función en Haskell que dada una cadena de caracteres devuelva el número de consonantes que hay en dicha cadena. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> contarConsonantes "Ejercicio 3. Programacion funcional, enero 2017"
18
```

10. Implementar una función en Haskell que dado un número entero n devuelva una lista que contenga los n primeros números primos de Mersenne. M es un número primo de Mersenne cuando se cumple lo siguiente:
    - M es un número primo
    - M es un número de Mersenne, es decir, un número una unidad menor que una potencia entera positiva de 2, M = 2p-1, siendo p un número primo positivo.

    Se debe utilizar la definición de listas por comprensión. Ejemplos de aplicación de la función podrían ser:

```
> listaMersenne 3
[3,7,31]
> listaMersenne 5
[3,7,31,127,2047]
```

11. Implementar una función en Haskel que determine si dos listas de enteros son iguales. Se considera que dos listas son iguales si contienen los mismos elementos, en el mismo orden. Debes utilizar la definición de listas por comprensión. Ejemplos de aplicación de la función podrían ser:

```
> listasIguales [1,2,3] [3,2,1]
False
> listasIguales [1,2,3] [1,2,3]
True
```