

Tema 2 – Listado I de ejercicios

1. Implementar una función que dados tres números enteros devuelva el mayor de ellos. Definir la función con guardas. Algunos ejemplos de aplicación de la función son:

```
> maxTres 5 8 2  
8
```

```
> maxTres 0 1 2  
2
```

2. Definir una función que dados tres números enteros devuelva el mayor de ellos. Se debe utilizar en la definición de esta función otra función que calcule el máximo de dos números enteros.
3. Implementar una función en Haskell que dados tres números enteros determine si están ordenados de menor a mayor. Ejemplos de aplicación de la función podrían ser:

```
> ordenadosMenor 3 4 5  
True  
> ordenadosMenor 4 3 5  
False
```

4. Implementar una función en Haskell que dada una tupla compuesta por tres números enteros la devuelva ordenada de menor a mayor. Un ejemplo de aplicación de la función podría ser los siguientes:

```
> ordenarTupla (5,1,4)  
(1,4,5)
```

5. Implementar una función que dado un número entero devuelva una lista que contenga todos los divisores de dicho número. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser:

```
> divisores 21  
[1,3,7,21]
```

6. Implementar una función que dado un carácter devuelva True si dicho carácter es un dígito

- a. Debes utilizar Guardas
 - b. Debes utilizar ajuste de patrones
- Un ejemplo de aplicación podría ser

```
> esDigito 'a'  
False  
> esDigito '9'  
True
```

7. Implementar una función que diga si un número es primo. Para ello se debe utilizar la función que calcula los divisores de un número (ejercicio 5). Ejemplos de aplicación de la función podrían ser

```
> esPrimo 5
True
> esPrimo 25
False
```

8. Implementar una función que dada una lista de enteros devuelva otra lista que contenga los elementos de la primera que son impares y números primos. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> listaPrimosImpares [1,2,3,5,4,6,9,15]
[1,3,5]
```

9. Implementar una función que dado un entero devuelva una lista con los números primos menores o iguales que dicho número. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> primosMenorIgual 17
[1,2,3,5,7,11,13,17]
```

10. Se quiere implementar una función que sea capaz de descifrar mensajes de texto que se han cifrado utilizando listas de 2-tuplas de caracteres. A la hora de descifrar el mensaje, el algoritmo funciona de la siguiente forma: dada la 2-tupla (c_1 , c_2), si c_2 es una vocal, c_1 formará parte del mensaje original. En cualquier otro caso c_1 será descartado. Se pide, implementar una función en Haskell que, dada una lista de 2-tuplas de caracteres (mensaje cifrado), devuelva el mensaje de texto descifrado. Debes utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> codificacionTuplas [('c','f'), ('a','e'), ('d','a'), ('e','c'), ('n','i'), ('a','j')]
"adn"
```

11. Implementar una función que dado un número entero n y una lista de 2-tuplas de enteros, devuelva una lista con las tuplas cuyo primer elemento es impar y mayor que n . Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser:

```
> filtrarTuplas [(1,2), (2,1), (4,5), (3,3), (7,2)] 2
[(3,3),(7,2)]
```

12. Implementar una función en Haskell que reciba una lista de ternas de números enteros (tres-tuplas) y nos devuelva el número de estas que son pitagóricas. Se entiende por terna pitagórica (x,y,z) aquella que cumple que $x^2 + y^2 = z^2$. Se debe

utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser:

```
> cuantasPitagoricas [(3,4,5),(-4,3,7),(3,-4,-5),(6,8,10),(7,3,9)]  
3
```

13. Implementar una función que dado un `String` cambie las mayúsculas por minúsculas y viceversa. Puedes emplear la función creada en el ejercicio 13. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> mayusculasMinusculas "CadenaPrueba"  
"cADENApRUEBA"
```

14. Implementar una función en Haskell que, dada una lista de enteros, devuelva un mensaje indicando el primer elemento de la lista y su longitud. Un ejemplo de aplicación de la función podría ser

```
> mensajeLista [1,2,4,5,6]  
"Primer elemento: 1, longitud: 5"
```

15. Implementar una función que diga cuántos caracteres en mayúscula están contenidos en una frase dada. Se debe utilizar la definición de listas por comprensión. Un ejemplo de aplicación de la función podría ser

```
> contarMayusculas " Ejercicios de PF"  
3
```