

INFORMATICA.

EXAMEN FINAL Junio 2004.

Haskell

Apellidos, Nombre

Grupo

NOTA: Indica también el tipo de todas las funciones que defines en los distintos ejercicios.

1) Define mediante una función de plegado las siguientes funciones:

- a) (2 *puntos*) Una función polimórfica `eliminarDuplicados` que devuelva la lista que se obtiene al eliminar los elementos duplicados de una lista, por ejemplo:

```
eliminarDuplicados [1,2,1,2,3,2,4] => [1,2,3,4]
```

- b) (2 *puntos*) Una función `paresImpares` que tome como argumento una lista de enteros y devuelva una tupla con dos listas. En la primera aparecerán los elementos pares del argumento y en la segunda los impares, por ejemplo:

```
paresImpares [10,12,13,14,4,7] => ([10,12,14,4], [13,7])
```

2) La **conjetura de Goldbach** es la siguiente:

Todo número par mayor que 2 puede escribirse como suma de dos números primos. (Se puede emplear dos veces el mismo número primo).

- a) (1.75 *puntos*) Para comprobar la conjetura, escribe una función `goldbach` que tome un número par mayor que 2 y devuelva una lista con los pares de números primos tales que su suma igualan al número leído. Por ejemplo:

```
goldbach 58 => [(5,53), (11,47), (17,41), (29,29)]
```

- b) (1.25 *puntos*) Escribe una función `cumplenGoldbach` que tome una lista de números pares mayor que 2 y devuelva `True` si todos cumplen la conjetura.

3) (3 *puntos*) Sean los siguientes tipos para representar matrices como listas de vectores (uno por cada fila de la matriz):

```
type Vector = [ Float ]  
type Matriz = [ Vector ]
```

```
m :: Matriz  
m = [ [1,2,3]  
      , [4,5,6]  
      , [7,8,9] ]
```

Define un operador `<*>` que calcule el producto de dos matrices.

INFORMATICA.

EXAMEN FINAL Junio 2004.

Pascal

Apellidos, Nombre

Grupo

1) (5 *puntos*) Disponemos de un fichero de texto que contiene varias líneas (cada una de 100 caracteres como máximo). Escribe un programa que, para dicho fichero, indique si el texto que aparece en cada línea es un identificador válido para el lenguaje Pascal.

Por ejemplo, para el siguiente fichero:

```
il
casa
josé
la variable
```

La salida del programa por pantalla deberá ser:

```
El texto de la línea 1 es un identificador
El texto de la línea 2 es un identificador
El texto de la línea 3 NO es un identificador
El texto de la línea 4 NO es un identificador
```

2) Consideremos los siguientes tipos para representar los datos relativos a personas:

```
CONST
    MAX = 100;
TYPE
    PERSONA = RECORD
        Nombre, Apellido : String;
        Edad : Cardinal
    END;
    VECTOR = ARRAY [1..MAX] OF PERSONA;
```

- a) (2.5 *puntos*) Escribe un subprograma que tome como parámetro un VECTOR y lo ordene usando el método de **ordenación por selección** de modo **DESCENDENTE**, es decir de mayor a menor, tomando como criterio de ordenación los apellidos de las personas.
- b) (2.5 *puntos*) Escribe una función que tome como parámetro un VECTOR de personas ordenado **DESCENDENTE** y un apellido, y realice una búsqueda binaria dentro del vector. Si se encuentra el apellido, la función debe devolver la posición de éste dentro del vector. Si no se encuentra el apellido, la función debe devolver cero.