Todas las preguntas tienen la misma ponderación (25%).
Se debe entregar esta hoja con el examen.

1. Tenemos una especificación formal axiomática para listas (**L**) de números naturales (**N**) con las siguientes operaciones, usando la sintaxis de Maude:

Sintaxis	**Descripción**
`listaVacia : -> L .`	Constante que representa la lista vacía.
`insertar : N L -> L .`	Devuelve la lista resultante de añadir al principio del segundo parámetro el natural representado por el primer parámetro.
`esMenorIgual : N N -> Bool .`	Comparación de naturales. Devuelve *true* si el primer parámetro es menor o igual que el segundo.

Estudiar y determinar el significado de la operación `misterio` cuya especificación se describe a continuación. Esta operación se apoya en otras dos operaciones, `auxiliar1` y `auxiliar2`, cuya especificación también se facilita.

Sintaxis:

```
op misterio : L -> L .
op auxiliar1 : L -> Bool .
op auxiliar2 : L -> L .
```

Axiomas:

```
var n, m : N .
var l : L .
eq misterio (l) = if auxiliar1 (l) then
                        l
                  else
                        misterio ( auxiliar2 (l))
                  fi .

eq auxiliar1 (listaVacia) = true .
eq auxiliar1 (insertar (n, listaVacia)) = true .
eq auxiliar1 (insertar (n, insertar (m, l))) =
       esMenorIgual (n, m) and auxiliar1 (insertar (m, l)) .

eq auxiliar2 (listaVacia) = listaVacia .
eq auxiliar2 (insertar (n, listaVacia)) = insertar (n, listaVacia) .
eq auxiliar2 (insertar (n, insertar (m, l))) =
                if esMenorIgual (n, m) then
                      insertar (n, auxiliar2 (insertar (m, l)))
                else
                      insertar (m, auxiliar2 (insertar (n, l)))
                fi .
```

Hay que describir de forma razonada qué es lo que hacen estas tres operaciones y mostrar algún ejemplo de su funcionamiento.

Todas las preguntas tienen la misma ponderación (25%).
No se debe entregar esta hoja (te la puedes quedar).

2. Tenemos que representar un tipo de datos *polinomio escaso*, que serán polinomios de grado potencialmente muy alto (hasta x^{10000}), de la forma:

$$P(x) = \sum_{i=0}^{10000} a_i \cdot x^i$$

El término "escaso" significa que la mayor parte de los términos valen 0. Por ejemplo, un polinomio escaso podría ser: $p1(x) = 4{,}5 \cdot x^{37} + 3{,}71 \cdot x^{198} + 0{,}37 \cdot x^{2500} - 1{,}62 \cdot x^{3030}$. O, por ejemplo: $p2(x) = 3{,}25 \cdot x^{92} - 9 \cdot x^{5241} + 8{,}08 \cdot x^{632}$. En concreto, no tendrán más de 50 coeficientes distintos de cero. Las operaciones que se espera realizar con estos polinomios son:

Sintaxis	Descripción
op polinomioNulo () : Polinomio	Operación constante que devuelve el polinomio p(x) = 0.
op añadir_coef (p: Polinomio, i: entero, a: real) : Polinomio	Esta operación devuelve el resultado de añadir al polinomio p un nuevo término $a \cdot x^i$.
op eval (p: Polinomio, x: real) : real	Devuelve el resultado de evaluar el polinomio p en x, es decir, devuelve el valor de p(x).

Describir de forma completa, detallada y justificada cómo se podrían utilizar las tablas de dispersión para representar este tipo de datos. Esta descripción debe contener:

- La definición del tipo de datos "polinomio escaso" utilizando la representación propuesta con tablas de dispersión.
- Las decisiones de diseño de las tablas de dispersión usadas: tipo de tablas, función de dispersión, tamaño de tabla, estrategia de reestructuración, etc.
- Mostrar gráficamente algún ejemplo de estas tablas (p.ej. usando los dos ejemplos de arriba).
- La implementación en pseudocódigo de las tres operaciones, explicando su funcionamiento.
- El estudio de la eficiencia computacional, en tiempo de ejecución, de las operaciones.

3. En una aplicación tenemos almacenado un conjunto de palabras utilizando una estructura de árboles. Escribir una operación que liste de forma eficiente todas las palabras que se encuentren alfabéticamente entre dos palabras dadas **pmin** y **pmax** (ambas inclusive). Por ejemplo, podemos consultar las palabras que están entre "caballo" y "cabello"; o entre "fuerza" y "mano".
Para hacer este ejercicio, puedes elegir y utilizar cualquiera de las estructuras de árboles vistas en clase. Si se hace con árboles trie, no hay que acceder a los nodos, sino utilizar las operaciones genéricas sobre nodos trie: Consulta (n: NodoTrie, c: carácter): NodoTrie, y el iterador para cada carácter c hijo del nodo n hacer.
Desarrollar la solución, escribiendo el pseudocódigo del algoritmo, explicando la solución, mostrando algún ejemplo y haciendo una estimación del tiempo de ejecución del algoritmo.

4. Este año la gripe se está extendiendo a una velocidad alarmante. El contagio se produce cuando dos personas, una enferma y otra sana, se encuentran; a partir del siguiente día, la persona que estaba sana pasa a estar enferma y puede contagiar a otros. Suponer que en cierta población tenemos **n** personas. Tenemos una matriz **M** de booleanos de tamaño **n**×**n** que indica en cada posición **M**[**i**, **j**] si las personas **i** y **j** se encuentran diariamente; esta matriz es simétrica. Por otro lado, el array **E** de booleanos de tamaño **n**, indica en cada **E**[**i**] si la persona **i** está inicialmente enferma el día 1 o no.
Escribir un algoritmo completo y eficiente que calcule tres cosas: (1) el número de personas que se acaban contagiando; (2) el día en el cual ya no se producen más contagios; y (3) las personas que no se contagiarán (porque no tienen ninguna relación directa o indirecta con otros contagiados). Escribir el algoritmo en pseudocódigo y explicar la solución.