Universidad de Las Palmas de Gran Canaria
Escuela de Ingeniería Informática
Examen de Programación IV. Parcial de Programación Funcional y Lógica
11 de mayo de 2015

Nombre y apellidos: .....

Notas:

- La duración del examen es de 60 minutos.
- Permanezca en su lugar. Para hablar con el profesor, pida permiso para levantarse.
- Debe tener visible sobre la mesa un documento que le identifique.
- No se permite el acceso a otra información que no sea este enunciado.
- El examen se realizará utilizando únicamente la página de edición correspondiente en el servidor aulaga.
- Sólo se permite un lápiz o bolígrafo durante la realización el examen.
- Si necesita escribir anotaciones, use el reverso de esta hoja, que deberá entregar al finalizar.
- Antes de comenzar el examen escriba su nombre arriba.

1) (2 punto) Defina en Scheme una función de nombre **`principio`** a la que se le pasa una lista y devuelve una lista con los elementos iguales al principio de la lista (vacía para una lista vacía).
Ejemplos:
`(principio '(a a a c a d a e f b))` *devuelve:* **`'(a a a)`**
`(principio '(a c c d a e f b))` *devuelve:* **`'(a)`**

2) (3 puntos) Defina una función de nombre **`profundidad`** a la cual se le pasa como primer parámetro una lista que puede contener átomos y listas recursivamente (ver ejemplos) y como segundo parámetro un átomo. Los valores de los átomos no se repiten en toda la estructura. La función devuelve a qué nivel de profundidad se localiza el átomo en la lista o cero si no se encuentra. Los niveles de profundidad se miden como 1 para el primero, 2 segundo, etc. Se recuerda que existe la función **list?.**
Ejemplos:
`(profundidad '(c a b (r t) f (1 ((3 4) 5) a)` *devuelve:* **`1`**
`(profundidad '(c a b (r t) f (1 ((3 4) 5) k)` *devuelve:* **`0`**
`(profundidad '(c a b (r t) f (1 ((3 4) 5) t)` *devuelve:* **`2`**
`(profundidad '(c a b (r t) f (1 ((3 4) 5) 4)` *devuelve:* **`4`**

3) (2 punto) Defina en Prolog un predicado **`principio(`***L, P***`)`** tal que, dada una lista *L*, calcule una lista *P* con los elementos iguales al principio de la lista *L*. Si la lista está vacía *P* será la lista vacía.
Ejemplo:
`?- cuenta([a, a, a, c, a, d, a, e, f, b], N).`
responde:
`N = [a, a, a]`

4) (3 puntos) Defina un predicado **`profundidad(`***L, A, N***`)`** tal que, dado una lista *L* que puede contener átomos y listas recursivamente (ver ejemplos), y un átomo *A*, *N* toma el valor de la profundidad donde se encuentre *A* en *L* o cero si no se encuentra. Los valores de los átomos no se repiten en toda la estructura. Los niveles de profundidad se miden como 1 para el primero, 2 segundo, etc.
Ejemplo:
`?- profundidad([c, a, b, [r, t], f, [1, [[3, 4], 5]], q], a, N).`
responde: **`N = 1`**
`?- profundidad([c, a, b, [r, t], f, [1, [[3, 4], 5]], q], k, N).`
responde: **`N = 0`**
`?- profundidad([c, a, b, [r, t], f, [1, [[3, 4], 5]], q], t, N).`
responde: **`N = 2`**
`?- profundidad([c, a, b, [r, t], f, [1, [[3, 4], 5]], q], 4, N).`
responde: **`N = 4`**