**Programación II, 2015-2016**

**Escuela Politécnica Superior, UAM**

**Práctica 1: Estructuras de Datos y Tipos Abstractos de Datos**

PARTE 2: PREGUNTAS SOBRE LA PRÁCTICA

1. ¿Sería posible implementar la función de copia de puntos empleando el siguiente prototipo  
    **STATUS node\_copy(Node pDest, const Node pOrigin);** ? ¿Por qué?

*No sería posible ya que como argumentos se deberían pasar punteros a Nodo, ya que si esto no*

*se hace, pDest y pOrigin sin variables locales de node\_copy, y al llamar la función, ésta no haría*

*ningún cambio a los nodos pasados como argumento.*

1. ¿Es imprescindible el puntero Node\* en **int node\_print(FILE \* pf, const Node\* p);** o podría ser **int node\_print(FILE \* pf, const Node p); ?**

Si la respuesta es sí: ¿Por qué?  
Si la respuesta es no: ¿Por qué se utiliza, entonces?

*El puntero es imprescindible ya que el TAD Node está definido fuera del fichero .c, en el node.h.*

1. ¿Qué cambios habría que hacer en la función de copiar nodos si quisiéramos que recibiera un nodo como argumento donde hubiera que copiar la información? Es decir, ¿cómo se tendría que implementar si en lugar de **Node\* node\_copy(const Node\* nOrigin)**, se hubiera definido como **STATUS node\_copy(const Node\* nSource, Node\* nDest)**?¿Lo siguiente sería válido: **STATUS node\_copy(const Node\* nSource, Node\*\* nDest)**? Discute las diferencias.

*La sentencia “STATUS node\_copy(const Node\* nSource, Node\*\* nDest” sería correcta ya que al ser una función Status, no se puede devolver el nodo obtenido con un return. Por ello se usa el doble puntero, el cual almacena la dirección de una variable que apunta a Node. Por todo ello, cuando se modifica el nodo nDest dentro de la función, se modifica también el nodo de fuera de ella cuya dirección ha sido pasada como argumento en la función.*

1. Indica qué se tendría que cambiar en **graph.c/h** para tener grafos que pudieran almacenar cualquier estructura de datos, es decir, que no estuviera limitado a almacenar nodos definidos como el TAD Node.

*La única manera que se nos ocurre es ampliar el TAD de graph, creando un apartado para cada tipo de estructura que se quiera almacenar. Después, habría que crear nuevas funciones para poder trabajar con los diferentes tipos de datos, ya que las funciones a las que se pasa algún elemento como argumento no pueden ser genéricas, es decir no puedes pasar cualquier tipo de variable.*