



Universidad  
Carlos III de Madrid

## Programación de Sistemas Grado en Ingeniería Telemática

Leganés, 24 de abril de 2013  
Duración de la prueba: 15 min

Examen parcial 2 (teoría)  
Puntuación: 5 puntos sobre 10 del examen

*Sólo una opción es correcta en cada pregunta de opción múltiple. Cada respuesta correcta suma 1/2 puntos. Cada respuesta incorrecta resta 1/6 puntos. Las preguntas no contestadas no suman ni restan puntos.*

- Marca la respuesta a cada pregunta con una equis (“X”) en la tabla de abajo.
- Si marcas más de una opción, o ninguna opción, la pregunta se cuenta como no contestada (ni suma ni resta).
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Nombre:

Grupo:

Firma:

NIA:

--	--	--	--	--	--	--	--	--	--

	A	B	C	D		A	B	C	D
1					6				
2					7				
3					8				
4					9				
5					10				

■   ■   ■   ■   ■   ■   ■   ■   ■   ■

1.- El método de la derecha:

- (a) No tiene caso recursivo.
- (b) \*\*\* Devuelve el resultado de restar b a a, si b es mayor o igual que cero.
- (c) No tiene caso base.
- (d) Devuelve el resultado de sumar a y b, si b es mayor o igual que cero.

```
public int method(int a, int b) {  
    if (b == 0) {  
        return a;  
    } else {  
        return method(a - 1, b - 1);  
    }  
}
```

2.- Indica cuál de las siguientes afirmaciones es cierta:

- (a) \*\*\* En un método recursivo lineal hay sólo una llamada recursiva en cada caso recursivo.
- (b) La implementación de un método mediante recursión en cascada es siempre preferible a su implementación mediante recursión lineal.
- (c) Un método recursivo no por la cola puede siempre transformarse de forma inmediata y sistemática en un método iterativo.
- (d) La recursión mutua es un caso especial de recursión anidada.

3.- En el último nodo de una lista enlazada, el atributo que almacena la referencia al siguiente nodo:

- (a) \*\*\* Tiene valor *null*.
- (b) Apunta al objeto guardado en el nodo.
- (c) Apunta al primer nodo de la lista.
- (d) Apunta al penúltimo nodo de la lista.

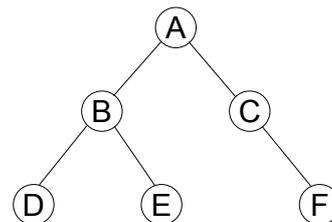
4.- ¿Qué instrucción debe aparecer en vez de "XXXXX" en la siguiente secuencia de código que trabaja sobre una pila para que a la variable `result` se le asigne el valor "D"?

- (a) `stack.pop()`
- (b) \*\*\* `stack.push("A")`
- (c) `stack.top()`
- (d) `stack.size()`

```
Stack stack = new Stack();  
stack.push("A");  
stack.push("B");  
stack.push("C");  
stack.pop();  
stack.push("D");  
XXXXX  
stack.pop();  
Object result = stack.pop();
```

5.- Si se recorre el siguiente árbol en postorden, ¿en qué orden se procesan los nodos?

- (a) \*\*\* D, E, B, F, C, A.
- (b) B, D, E, C, F, A.
- (c) A, B, D, E, C, F.
- (d) D, E, B, A, C, F.



6.- Indica cuál de las siguientes afirmaciones es *falsa*:

- (a) Una pila es una estructura de datos lineal.
- (b) \*\*\* En una pila implementada mediante lista enlazada se inserta por un extremo y se extrae por el contrario.
- (c) En una pila implementada mediante lista enlazada es más conveniente extraer por el principio de la misma.
- (d) Una pila implementada mediante *array* tiene una capacidad máxima que se establece en el momento en que se crea.

7.- El siguiente método de la clase BNode:

- ```
public boolean method(Object info) {
    boolean result;
    if (this.info.equals(info)) {
        result = true;
    } else if (left != null && left.method(info)) {
        result = true;
    } else if (right != null && right.method(info)) {
        result = true;
    } else {
        result = false;
    }
    return result;
}
```
- (a) Devuelve *true* si y sólo si el dato en la raíz del árbol es igual al recibido como parámetro.
  - (b) \*\*\* Devuelve *true* si y sólo si en el árbol hay al menos un nodo con un dato igual al recibido como parámetro.
  - (c) Devuelve *true* si el árbol tiene un número par de nodos.
  - (d) Recorre el árbol en postorden.

8.- ¿Cuál de los siguientes fragmentos de código encola un nuevo dato en una cola?

- (a) 

```
public void enqueue(Object info) {
    Node n = new Node(info, first);
    if (top == null) {
        n = top;
    } else {
        tail.setNext(n);
    }
    tail = n;
}
```
- (b) 

```
public void enqueue(Object info) {
    Node n = new Node(info, first);
    tail.setNext(n);
    tail = n;
}
```
- (c) 

```
public void enqueue(Object info) {
    Node n = new Node(info, null);
    tail.setNext(n);
    tail = n;
}
```

(d) \*\*\*

```
public void enqueue(Object info) {
    Node n = new Node(info, null);
    if (top == null) {
        top = n;
    } else {
        tail.setNext(n);
    }
    tail = n;
}
```

9.- El siguiente fragmento de código en una lista enlazada:

```
Node n = new Node(info);
n.setNext(first);
first = n;
```

- (a) Inserta un nuevo nodo al final de la lista.
- (b) Inserta un nuevo nodo en una posición intermedia de la lista, inmediatamente antes del nodo al cual apunte *first*.
- (c) Inserta un nuevo nodo en una posición intermedia de la lista, inmediatamente a continuación del nodo al cual apunte *first*.
- (d) \*\*\* Inserta un nuevo nodo al principio de la lista.

10.- ¿Qué tipo de recursión se utiliza en el siguiente método?:

- (a) Lineal no por la cola.
- (b) Mutua.
- (c) \*\*\* En cascada.
- (d) Anidada.

```
public double s(int n) {
    if (n <= 1) {
        return 1.0 + n;
    } else {
        return 0.5 * s(n - 1) + 0.1 * s(n - 2);
    }
}
```