



Universidad  
Carlos III de Madrid

## Systems Programming Groups 65-69-79 and 95

Leganés, May 9, 2014  
Duration: 10 min

Mid-Term Exam 2 (test)  
Score: 5 points over 10

*Only one answer is correct for each question. Each correct answer adds  $\frac{1}{2}$  points. Each wrong answer subtracts  $\frac{1}{6}$  points. Not answered questions do not add nor subtract points.*

- No books or notes are allowed. Also, mobile phones and any other electronic devices must be off. Failure to comply with any of these norms can be reason enough for immediate expulsion out of the exam.
- Mark the answer to each question with an “X” in the table below.
- If none or more than one option is marked, the question is considered as not answered (thus not adding or subtracting points).
- Fill in your **personal details** before you start the test.

**Model: A**

Name:

Group:

Signature:

NIA:

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

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

■ ■ ■                      ■ ■ ■

1.- For which integer values of  $n$  is the following method valid?

```
public int calculus(int n) {
    if (n <= 1)
        return 1;
    else
        return calculus(n - 1) + n;
}
```

- (a) For  $n \leq 1$ .
- (b) For  $n < 1$ .
- (c) \*\*\* For all  $n$ .
- (d) For  $n \geq 1$ .

2.- In a binary search tree, searches are performed by:

- (a) Traversing the tree in pre-order.
- (b) Traversing the tree in post-order.
- (c) Comparing the keys of the current node among them.
- (d) \*\*\* Comparing the target key against the key of the current node.

3.- The following method:

```
void test(Stack s) {
    Queue q = new Queue();
    while (!s.isEmpty())
        q.enqueue(s.pop());
    while (!q.isEmpty())
        s.push(q.dequeue());
}
```

- (a) Inverts the order of the elements of the queue.
- (b) Leaves both the stack and the queue unchanged.
- (c) Inverts the order of the elements of the stack, but the change is not visible outside the method because method parameters are passed by value in Java.
- (d) \*\*\* Inverts the order of the elements of the stack.

4.- Which methods of a double queue (*Deque*) would be used for implementing the *push* and *pop* methods of a stack?

- (a) *insertLast* for *push* and *removeFirst* for *pop*
- (b) *insertLast* for *pop* and *removeLast* for *push*
- (c) *insertFirst* for *push* and *removeLast* for *pop*
- (d) \*\*\* *insertFirst* for *push* and *removeFirst* for *pop*

5.- The code to remove the first node in a linked list is:

- (a) `*** top = top.getNext()`
- (b) `top.setNext(top.getNext())`
- (c) `setNext(top.getNext())`
- (d) `top = tail.getNext()`

6.- Given the following *array* of integers that represents a binary heap: {1, 4, 3, 5, 10, 7, 9, 8}, when element 1 is extracted, the resulting heap will be:

- (a) {3, 4, 5, 7, 8, 9, 10}
- (b) {4, 3, 5, 10, 7, 9, 8}
- (c) {1, 4, 3, 5, 10, 7, 9}.
- (d) `***` {3, 4, 7, 5, 10, 8, 9}

7.- Given a non-empty binary search tree, after the following piece of code is run, the reference *subtree*:

```
BSTree subtree = tree;
while (!isEmpty(subtree) && !isEmpty(subtree.getRight())) {
    subtree = subtree.getRight();
}
```

- (a) Always refers to a subtree different than the whole tree.
- (b) Always refers to the whole tree.
- (c) Always refers to an internal subtree.
- (d) `***` Always refers to the subtree whose root node stores the maximum key.

8.- Indicate what would return the following recursive method for  $n = 3$

```
public int calculus(int n) {
    if (n <= 0)
        return 1;
    else
        return 3 + calculus(n - 2);
}
```

- (a) `***` 7.
- (b) 4.
- (c) It never ends.
- (d) 6.

9.- Assume that the following sequence of operations is performed on a given queue: *enqueue(1)*, *enqueue(3)*, *enqueue(2)*, *dequeue()*, *enqueue(4)*. If the following operation is *dequeue()*, it returns:

- (a) 2.
- (b) 4.
- (c) \*\*\* 3.
- (d) 1.

10.- In general, in a tree:

- (a) \*\*\* There is a unique path from the root to each node.
- (b) Each node can have a maximum of 2 children.
- (c) There may be a maximum of 2 paths from one node to another.
- (d) The height of the tree is always less than its width.