

Being A a binary search tree that has been filled using the following sequence of nodes: 9, 4, 1, 8, 0, 7, 2, 3, 6, 5, what is the path inorder?

Select one:

- ☐ a. 0,1,3,2,4,5,6,7,8,9
- ☐ b. 9,4,1,0,2,3,8,7,6,5
- ☐ c. 0,3,2,1,5,6,7,8,4,9
- ☒ d. 0,1,2,3,4,5,6,7,8,9



The insert method is used to insert an integer into an array of integers sorted in increasing order so that if

```
int[] array = {1,3,5,6,9,11};
```

executes

```
array = insert (7, array);
```

returns {1,3,5,6,7,9,11};

Mark which is the valid option that should replace the line ***something\_to\_do*** to perform the described function

```
static int[] insert(int val, int[] array){
    int n = array.length;
    int[] a2 = new int[n+1];
    int i;

    something_to_do
        a2[i] = array[i];
    a2[i] = val;
    for (;i<n;i++)
        a2[i+1]=array[i];"
    return(a2);
}
```

Select one:

- ☐ a. for(i=n; i>0; i--)
- ☒ b. for(i=0; i<n && array[i]<val; i++)
- ☐ c. for(i=0; array[i]>val; i--)
- ☐ d. for(i=n; i>0 && array[i]>val; i--)



What is the result of running m(10,1)?

```
static int m(int a, int b) {  
    if (a <= b)  
        return (1);  
    if (a % 2 == b % 2)  
        return (1 + m(a - 1, b + 1));  
    else  
        return (m(a, b - 1));  
}
```

Select one:

- ☒ a. 6
- ☐ b. 7
- ☐ c. 4
- ☐ d. 5



Being s a stack of strings, and q a queue of strings, both empty, indicate the value of x (of type String) when executing the following sequence of instructions:

```
s.push("A");  
q.enqueue(s.top());  
q.enqueue("B");  
s.push(q.dequeue());  
s.push("C");  
s.push(q.dequeue());  
x = s.pop()+s.pop()+s.pop();
```

Select one:

- ☐ a. AAC
- ☐ b. ABC
- ☐ c. CAnull
- ☒ d. BCA



Which methods of a deque would be used to implement the enqueue and dequeue methods of a queue?

Select one:

- ☒ a. insertFirst for enqueue and removeLast for dequeue
- ☐ b. removeFirst for enqueue and removeLast for dequeue
- ☐ c. insertLast for enqueue and removeLast for dequeue
- ☐ d. insertFirst for enqueue and removeFirst for dequeue



You need to do a black-box testing for a method that calculates the area of a square. This method received one parameter, the side, of type double. What set of values would be used to do the black-box testing of this method considering all equivalence classes and boundary values?

Select one:

- ☐ a. -1.2, -1.8, -0.7, 2.35
- ☐ b. 0.0, 1.0, 2.0
- ☐ c. 0.0, 0.1, 0.2, 0.3, 0.4, 0.5
- ☒ d. 2.1, 0.0, -1.75



Which of the following is true about the m method?

```
static int m(int a, int b) {  
    if (a < b)  
        return (m (b,a));  
    if (b < 0)  
        return (a);  
  
    return (m(2*a, b - 1));  
}
```

Select one:

- ☐ a. It has two base cases (a<b) and (b<0)
- ☐ b. It is a cascading tail recursion
- ☐ c. It is a cascading not tail recursion
- ☒ d. It is a linear tail recursion



The modifier final applied on the parameter a in the following method...

```
public int m(final int a){...}
```

Select one:

- ☒ a. Indicates that the value taken by the parameter a when calling the method m cannot be modified inside that method.
- ☐ b. Indicates that the method m cannot return a, so the return a; statement would give a compilation error.
- ☐ c. Indicates that the method m cannot be overridden in the subclasses of the class where it is implemented.
- ☐ d. Indicates that the memory space occupied by the parameter a will not be released when finishing the execution of method m.



We wanted to make a variation of the *BubbleSort*, so that the lowest values are placed in the lowest positions first, but we are missing the line ***I\_dont\_know\_what***.

```
static void sort2 (int[] array) {  
    int n = array.length;  
    for (int i=0; i<n; i++)  
        i_dont_know_what  
        if(array[j]<array[j+1])  
            swap(array, j, j+1);  
}
```

Which of the alternatives is correct?

Select one:

- ☒ a. for (int j=n-2; j>=i; j--)
- ☐ b. for (int j=n; j>=i; j--)
- ☐ c. for (int j=n-1; j>i; j--)
- ☐ d. for (int j=n; j>i; j--)



Given the following code and the test class, what line coverage is achieved?

```
public class C {
    public static int m(int a){
        if (a>0){
            return 1;
        } else if (a<0){
            return -1;
        } else{
            return 0;
        }
    }
}

public class CTest {
    C c = new C();
    @Test
    public void test1() {
        assertEquals(C.m(-5), -1);
    }
    @Test
    public void test2() {
        assertEquals(C.m(0), 0);
    }
    @Test
    public void test3() {
        assertEquals(C.m(10), 1);
    }
}
```

Select one:

- ☒ a. 100%
- ☐ b. 66.6%
- ☐ c. 50%
- ☐ d. 33.3%



In a linked list we implement the recursive method **m**.

```
static int m(Node<E>a, Node<E>b) {
    if (b == null)
        return (a);
    return (m(b, b.getNext()));
}
```

What do you get when calling **m(null, first);**?

Select one:

- ☐ a. NullPointerException
- ☐ b. null
- ☒ c. The last node in the list
- ☐ d. The first node in the list



Given the following code, what does the program print on screen?

```
public class C{
    public static void main (String[] args){
        int a = 0;
        m1(a);
        m2(a);
        System.out.println(a);
    }
    public static void m1(int a){
        ++a;
    }
    public static void m2(int a){
        a++;
    }
}
```

Select one:

- ☐ a. 2
- ☐ b. 1
- ☐ c. a
- ☒ d. 0



For an algorithm to be  $O(n/2)$  means that:

Select one:

- ☐ a. That the computation time grows as  $n/2$ .
- ☒ b. This does not exist.
- ☐ c. That consumes more resources as  $n/2$  tends to infinity
- ☐ d. That the additional memory it uses is half the size of the data



If you have the attribute protected int i in class A and you implement class B that extends class A:

Select one:


- ☒ a. It is possible to modify directly the value in attribute i from B.
- ☐ b. It is possible to read the value in attribute i from B although the value in i cannot be modified from B.
- ☐ c. In order to read or modify the value in attribute i from B it is necessary to use super.
- ☐ d. It is only possible to read the value in attribute i from class A.



Given the following program with each class in a different file, which statement is correct?


```
public class A {
    public A(){}
    public void print(){
        System.out.print("printA ");
    }
}
public class B extends A{
    public B(){}
    public void print(){
        System.out.print("printB ");
    }
}
public class C extends A {
    public C(){}
    public void print(){
        System.out.print("printC ");
    }
}
public class Z {
    public static void main(String[] args) {
        A[] array = {new A(), new B(), new C()};
        for (int i=0; i<3; i++){
            array[i].print();
        }
    }
}
```

Select one:

- ☐ a. The output of the program is: printA printA printA.
- ☒ b. The output of the program is: printA printB printC. 
- ☐ c. There is an error in class Z due to incompatible data types.
- ☐ d. The for loop takes advantage of method overloading.


Given the statement `public class A extends B implements C { ... }` it can be said that...

Select one:

- ☐ a. A and C are classes, B is an interface
- ☐ b. A, B, and C are all classes
- ☐ c. A is a class, B and C are interfaces
- ☒ d. A and B are classes and C is an interface 


Select the INCORRECT option regarding abstract classes in Java:

Select one:

- ☐ a. An abstract class cannot be instantiated
- ☒ b. An abstract class cannot be extended 
- ☐ c. An abstract class must provide at least one constructor and if there is none explicitly programmed, a default constructor with no arguments is used.
- ☐ d. An abstract class can only extend a parent class


A is a complete binary tree that has 256 leaves. What is its height?

Select one:

- ☐ a. 6
- ☒ b. 8 
- ☐ c. 7
- ☐ d. 5


Class B extends Class A and both have the method public String method toString(), which of the following statements is INCORRECT?

Select one:

- ☐ a. The toString method of B overrides the toString method of A
- ☐ b. Class A implicitly extends (like all java classes) the Object class and overrides its toString() method
- ☐ c. The toString method of B can invoke the toString method of A with super.toString()
- ☒ d. The code A a = new B(); System.out.println(a); calls the toString() implementation provided in class A 

The reserved word this is used in Java to refer to:

Select one:

- ☐ a. An attribute
- ☐ b. A method
- ☒ c. An object 
- ☐ d. A constructor