

1. Consider the following list of **doubles**, sorted in order. Perform a Binary Search for the indicated value, showing your work clearly by describing the variables. Determine the **number of steps (number of times mid is calculated)** and the **value returned**.

Search for 94.6

0	1	2	3	4	5	6	7	8	9	10	11	12
9.6	24.3	25.7	35.7	41.1	53.2	55.9	62.9	67.0	84.7	88.6	91.8	92.3

2. Consider the following list of **ints**, sorted in order. Perform a Binary Search for the indicated value, showing your work clearly by describing the variables. Determine the **number of steps (number of times mid is calculated)** and the **value returned**.

Search for 45

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
13	25	26	27	31	34	36	38	41	45	62	68	77	79	82

3. Consider an array of 1800 (sorted) **Strings**. Assume that we apply binary search algorithm to this array, looking for an element that exists in the array. What is the maximum number of steps that it can take to find this element?

4. Consider an array of 700 (sorted) **chars**. Assume that we apply a binary search algorithm to this array, looking for an element that does not exist in the array. What is the maximum number of steps that it can take to determine that this element does not exist in the list?

Problems 5 and 6. Given the following numbers in the given base, convert the number to the indicated bases.

5. Convert 1819_{10} (a decimal number) to both a binary (base 2) and octal (base 8) number

6. Convert $4DB_{16}$ (a hexadecimal number) to both a decimal (base 10) and binary (base 2) number

7. What is the output from the following:

```
int num1 = 4, num2 = 5;
Integer int1 = new Integer(12);
Integer int2 = new Integer(63);
swapper(num1, num2, int1, int2);
System.out.print(num1 + " " + num2 + " " + int1 + " " + int2 + " ");
```

where the method **swapper** is defined as follows:

```
public void swapper (int n1, int n2, Integer i1, Integer i2)
{
    int temp = n1;
    n1 = n2;
    n2 = temp;
    Integer tempint = i1;
    i1 = i2;
    i2 = tempint;
    System.out.print(n1 + " " + n2 + " " + i1 + " " + i2 + " ");
}
```

- (A) 5 4 63 12 5 4 63 12
- (B) 5 4 63 12 4 5 63 12
- (C) 5 4 63 12 4 5 12 63
- (D) 4 5 12 63 4 5 12 63
- (E) The method **swapper** will produce errors when it is compiled.

8. Consider the following data field and method.

```
private int[][] mat; // mat is square

public void mark(int rowMark, int colMark)
{
    for(int k = 0; k < mat.length; k++)
    {
        int row = (rowMark + k) % mat.length;
        int col = (colMark + mat.length - k) % mat.length;
        mat[row][col] = 1;
    }
}
```

Assume **mat** is a **5-by-5** array containing all zeroes. Which of the following shows **mat** after the call **mark(3, 1)**?

(A) 1 0 0 0 0	(B) 0 0 0 1 0	(C) 0 0 0 0 0	(D) 0 0 1 0 0	(E) 0 0 0 0 1
0 1 0 0 0	0 0 0 0 1	0 0 0 0 0	0 0 0 1 0	0 0 0 1 0
0 0 1 0 0	1 0 0 0 0	0 0 0 0 0	0 0 0 0 1	0 0 1 0 0
0 0 0 1 0	0 1 0 0 0	0 1 0 0 0	1 0 0 0 0	0 1 0 0 0
0 0 0 0 1	0 0 1 0 0	0 0 1 0 0	0 1 0 0 0	1 0 0 0 0

9. What keyword is used to specify that a data member is a **class** data member (shared among all instances of that class)?

(A) **final** (B) **shared** (C) **public** (D) **static** (E) **protected**

10. Which of the following is **TRUE**?

- (A) Multiple **catch** blocks should be listed in the order from specialized exception classes to more general ones.
- (B) If there is no exception, the **finally** block will not be executed.
- (C) If there are multiple **catch** blocks, the **finally** block will never be executed.
- (D) If there are multiple **catch** blocks, all blocks that match the exception will be executed.
- (E) Multiple **catch** blocks are not allowed in java.

11. A "worst case" situation for an insertion sort on 100 elements would be

- (A) A list sorted in reverse order
- (B) A list in random order
- (C) A list in correct sorted order
- (D) A list with the first half in sorted order, and the second half in random order
- (E) A list with the first 98 elements in random order, and the last 2 elements the smallest in the list

12. Which of the following messages passed to the **String str** could throw a **StringIndexOutOfBoundsException**?

- (A) **str.length();**
- (B) **str.charAt(2);**
- (C) **str.replace('a','A');**
- (D) **str.equals(str);**
- (E) any of the above could throw a **StringIndexOutOfBoundsException**

13. Consider the following class:

```
public class TestSample
{
    private ArrayList<Integer> samples;

    public TestSample (int n)
    {
        samples = new ArrayList<Integer>();
        for(int k = 0; k < n; k++)
        {
            samples.add(k);
        }
    }

    public double getBestRatio ( )
    {
        double maxRatio = samples.get(1).intValue() / samples.get(0).intValue();

        for(int k = 1; k < samples.size() - 1; k++)
        {
            double ratio = samples.get(k+1).intValue() / samples.get(k).intValue();
            if (ratio > maxRatio)
            {
                maxRatio = ratio;
            }
        }
        return maxRatio;
    }
}
```

What is the result of the following code segment?

```
TestSample test = new TestSample(2);    // notice that's a 2 there . . .
System.out.println(test.getBestRatio());
```

- (A) **NullPointerException**
- (B) **ArithmeticException**
- (C) **IndexOutOfBoundsException**
- (D) **ClassCastException**
- (E) **Infinity**

14. A list of numbers is stored in a sorted array. It is required that the list be maintained in sorted order. This requirement leads to inefficient execution for which of the following processes?

- I. Inserting and deleting numbers
- II. Finding the maximum value in the list
- III. Summing the five smallest numbers in the list

- (A) I only
- (B) II only
- (C) I and II only
- (D) I and III only
- (E) II and III only

