

Computer Science Contest #1314-02 Key

October 19, 2013

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

■

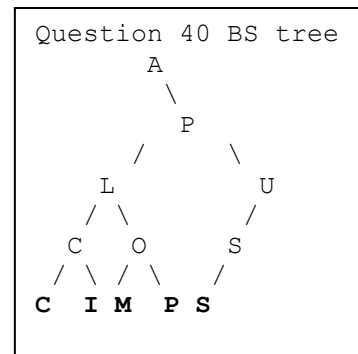
- 11) D
- 12) D
- 13) C
- 14) E
- 15) D
- 16) A
- 17) D
- 18) B
- 19) E
- 20) C

■

- 21) B
- 22) A
- 23) D
- 24) C
- 25) A
- 26) D
- 27) D
- 28) C
- 29) D
- 30) C

■

- 31) D
- 32) B
- 33) B
- 34) A
- 35) C
- 36) C
- 37) C
- 38) A
- 39) C

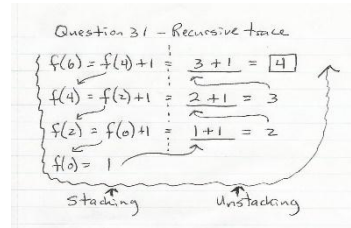


40) 5 leaves in the tree, shown in bold, and tree height is 4 (root node is at level zero, and furthest leaf is at level 4).

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Brief Explanations:



1. $1001_2 + 1011_2 = 9_{10} + 11_{10} = 20_{10} = 24_8 = 14_{16} = 10100_2$
2. $4 + 12 = 16$
3. Objects actually output "null" when assigned "null"
4. The loop starts at -3 and terminates at -9, with -7 being the last output
5. Remember that indexing for strings starts at 0, so "rain" is at position 3
6. This is simple...there are 5 elements in the list, so the length of the list is 5
7. "p&&q" means "p and q", which is only true when both are true. "p||q" means "p or q", true when either or both are true.
8. Indenting has no meaning in Java (unlike Python), so the second output statement is NOT attached to the if, and will always happen.
Therefore **w-10** will always output in this code.
9. The Math.cbrt method takes the cube root of a number, and the cube root of 27 is 3.0.
10. This is an asymmetrical matrix (rows are not all the same length). Using the length command gives you how many rows there are in all, in this case, 3, and using the length[1] command gives the number of elements in row 1 (remember that the top one is row 0), and so on...
11. In any class definition, you can tell the constructors since they have the same name as the class, and they don't have **void** or a **return type** in front like other methods do. You can have multiple constructors as long as the parameter signature is unique for each one. There are three constructors here..
12. The first two examples work since the default constructor (the one with no parameters) constructs a 6 string acoustic guitar. In the one parameter constructor, the use of the command **this()** calls the default constructor, and then assigns the value of the parameter to the numStrings field, resulting in the same thing. The third constructor does not work since the order of the parameters is reversed.
13. The output is controlled by the toString method definition.
14. When you have several assignment statements, the evaluation order is right to left, which means t is assigned first ($7.5 * 3 = 22.5$), and then that value is added to j ($5 + 22.5 = 27.5$). Since j is an int, the decimal is automatically truncated because of automatic casting with the += operator, resulting in 27 as the int answer.
15. This is simply the classic summation of the values 0-9, or $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$, which is 45.
16. The two parameter substring command starts at the first index position (the letter "f" in this case), and ends one place BEFORE the 2nd index position ("b"). One good way to check the answer is to subtract the index values ($12 - 6$), and that should be the length of the resulting string, "flyNeb", length of 6.
17. This nested loop tests all four combinations that can happen for two boolean values, represented by the integers 0 and 1 in this case. Using the & operator results in a "bitwise" operation exactly like "&&", or AND. Same thing with the "|", bitwise for OR.
18. The div and mod operators (/ and %) simply take the division and remainder answers from this integer being divided by 10. $34/10$ is 3, and $34\%10$ is 4.
19. The $<<2$ is equivalent to multiplying by 4 (2^2), and $>>3$ is dividing by 8 (2^3). Therefore 50 times 4 is 200, and 200 divided by 8 is 25.
20. This formatting question may seem very difficult, but you should know that the often used %f decimal format specifier **always** has six places after the decimal point, and that is the only one incorrect. You should become familiar with all of these formats, and there are even more with the printf method.
21. This is a formal parameter since it is in the method header. Actual parameters are in the client code in the method call, in this case, the b in the call **stuff(b)**.
22. <item1> is the return type in this case. You can tell it is a return type since it is called in an output statement in the client code, and that it is an **int** since it returns an **int**. Some may argue that **double** would also work here, but since the output examples for question 24 are integers, it is an **int**.
23. <item2> clearly must be the word **return**, since it is a method that returns a value, as indicated in question 22's explanation.
24. Since 14 is a multiple of 7, $14 * 7$, or 98 is returned. The second return statement is not reached. For the call with 15, the if statement is false, so the first return statement is ignored, and the second return statement is invoked, returning the value 15.
25. You need to learn this method...you'll see it often this year. The basic idea here is that it only returns significant digits, in this case, only 0, with no leading zeros.
26. This method simply returns how many bits are used to store a particular data type, in this case, 32. A long and a double use 64 bits, and a char 16.
27. This is simple unit circle stuff...the sine of a right angle is 1.
28. Here is left shift again, but this time by zero, which means nothing changes. $<<0$ means "times by 2^0 , or times by 1". The binary representation of 15 is 1111.
29. With ArrayLists, **add** simply puts something into the list, which outputs a **true** or **false** if the **add** was successful or not. The **get(2)** command returns the element in position 2 in the list, 9 in this case ([5, 7, 9])...don't forget zero indexing! The **size()** command simply returns the size of the list, 3 in this case since there are 3 elements in the list.
30. The Pattern class requires some study. Look it up and learn it! The "." means **any single character**. The ".*" pattern means any character, zero or more times. "+." means any character, one or more times. ".c." means any character, followed by the letter "c", then any character after that. The word "act" would match this pattern.
31. See a complete tracing of this recursive function call shown above.
32. The split pattern "[abc]" means split on any of those letters, wherever they occur in the string. There is a split on the "a" and on the "b", creating three elements in the resulting string array. The element in position 2 (the last position) is "lem".
33. The ternary operator is just like an **if else** statement...(w==42) is the if, (? "universal") is the result if true, and (: "answer") is the result if false, the **else**.
34. You will see more of these digital electronic symbols this year. This is the "bullet" shape, which means AND.
35. Simplifying Boolean expressions is important. This is an example of the "**absorption**" rule where if there are matching factors outside and inside the parentheses, the outside factor "absorbs" everything inside, in this case, the A absorbs everything inside.
36. The Arrays.sort in this case only sorts a subset of the list, namely between position zero and one place BEFORE position 4.
37. In a Priority Queue, **add** and **offer** both mean put something in the list in element order, **peek** means look at the element at the front, and **poll** and **remove** both remove the front element of the list.
38. In stacks, **push** means put it on the top of the stack, and **pop** means remove it from the top.
39. A merge sort process has a Big O rating of $O(N \log N)$. All of the rest are correct.
40. In this binary search tree exercise, duplicates are allowed and are inserted to the left of matching elements. The first node is at level zero, therefore the resulting tree, as you can see in the key above, has a height of 4, with 5 leaves.