Computer Science Answer Key
UIL Invitational B 2014

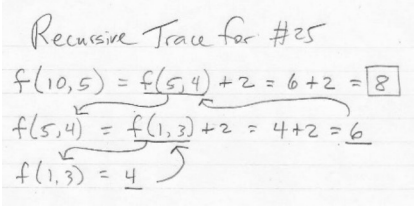
1) B	11) A	21) E	31) B
2) A	12) B	22) D	32) A
3) D	13) B	23) E	33) A
4) C	14) A	24) B	34) C
5) C	15) E	25) D	35) B
6) C	16) D	26) B	36) E
7) B	17) D	27) E	37) C
8) C	18) D	28) D	38) B
9) C	19) E	29) E	39) A
10) A	20) B	30) D	40) 5 last value popped 8 next to be popped

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. $110_2 + 100010_2 = 6_{10} + 34_{10} = 40_{10} = 50_8 = 28_{16} = 101000_2$
2. H = 24/5 = 4 (integer division)
3. Even though this is an array of Double objects, autoboxing **does not apply** when instantiating a static array like this. The 4 is an int and will cause a compile error, "incompatible types"
4. k starts at 3, outputs 6, 9 and 12, and stops at 12
5. the character 98 is the letter 'b', at position 5 from position 1. The 'b' in position 0 is not considered.
6. By the end of this assignment sequence, 4.5 is the element in every position.
7. p^q is p xor q, which requires opposites in order to be true, therefore p=true;p=false or p=false;q=true; will both evaluate to true.
8. The resulting values for all five choices are: "a",-1,"aa",0,"bb",3,"cccc",-1,"",-1
9. The minimum of -5.2 and 3.1 is -5.2
10. The 6 is in the second row (row 1), and in the third position of that row (column 2).
11. **`setNumStrings`** is a mutator method with a heading of **`public void.`**
12. It receives an integer **`(int n)`**
13. and assigns it to numStrings (**`numStrings = n;`**)
14. Since shift operations have priority over bitwise operations, 15 is left-shifted first, becoming 30, then 30 xor 30 is zero (Any value xor itself is zero – in assembly language that is one way of assigning a value of zero to a register).
15. The j values in this loop sequence are: 0, 0, 1, 3, 4, 12, 13, 39, 40, 120 and finally 121.
16. As is evident in the heading, this method is both a static method and a return method.
17. Since this is a chain if else, only one value is output for each call, according to the logic of the if statements. 9 produces 5, 8 produces 3, and 14 produces 5.
18. The (6,7) substring call is the correct one to access the letter "R".
19. This expression follows the order of operations, where 60%9 produces 6, and then is subtracted from 31 to make 25.
20. The Boolean expression is **p and q or q**, which when simplified just becomes **q** (Law of Absorption) and therefore each output digit matches the **q** digit of the term.
21. 28.5 mod 9 produces the value 1.5.
22. 180 degrees in radians is PI.
23. Any integer left shifted 32 spots (the bit size of the integer data type), will simply return to its original value. Essentially it is a Left Circle back to the original number. The Integer.toBinaryString method only outputs significant digits…no leading zeroes.
24. Since this the type of this ArrayList is not designated, any mixture of objects, including null, is acceptable. The final contents of the array are [null, 6, "ball", and 4.7].
25. See the recursive trace on the right for the solution to this problem.
26. The "`[pote]`" pattern splits at any of the letters 'p', 'o', 't' and 'e', which produces the array `["il","v","","","","ain"]` in this instance, producing "`vain`" for this output.
27. This ternary operation results in true, since 100%5 is equal to 0, therefore the resulting string is the one following the ?, which is "`walking`".
28. The first different characters in these two strings are 'r' and 'c', which produces 15 since 'r' is 15 places after 'c'.



29. Since hash structures guarantee no certain order, there is no indexing, therefore the call get(0) does NOT return the first element of the hash mapping, but simply looks for the mapping of the key value zero, and finding none, returns null.
30. Since the OR happens before the AND in this case, and AND occurs before OR in logic order, it necessary to use parentheses, producing (A OR B) AND C.
31. A ⊕ B simplifies to (not A and B or A and not B), which when FOILED with (A+B) produces the same thing, A ⊕ B.
32. The first "middle" found is the 6. Since 7 is to the right, the next "middle" is 9, then going left where the final "middle" is 7, the search item.
33. This structure is most certainly valid. Any class inheriting an abstract class is required to implement any abstract method in that class, so class B is REQUIRED to implement method two() from class A. Anything else is optional.
34. This is the product of 5 from method one, 2 from method two, and 2 from the variable x, for a result of 20.
35. The word "implements" is used when an interface is used, therefore option II is not valid. All of the rest are valid.
36. Since p.next.next pointer references the third node of the list, and the data for that node is 9, the resulting output is 9.
37. In this TreeSet process, 4 is added twice, but since there are no duplicates in sets, only instance remains. When the 4 is removed, leaving only the 5, 6, and 7, the size of the set is 3 and 6 is indeed in the list, resulting in the output **3true**.
38. Data input is a classic use of the try catch block. Since 3.14 works for doubles, no exception is thrown, and program flow drops to the finally block, which always executes, no matter what.
39. In the heapify process of a max heap, the process always starts at the bottom right of the tree, working left and upwards, switching any parent and child values that are not in correct max heap order. The first such occurrence here is the 6 and 7. Next will be the 2 and 9, and so on.
40. In this queue push and pop sequence, the 9 and 7 are pushed, then the 9 is popped, push the 5, 8, and 6, pop the 7 and the 5. The 5 was the last value popped, and the 8 sits at the front of the queue, waiting to be popped next.