Computer Science Contest #1415-14 Key

February 21, 2015

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

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

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

31) D
32) C
33) D
34) E
35) B
36) A
37) D
38) A
39) 19
40) W O R * L / - D +

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.

Explanations:

1. 96_{16} - 156_8 = 150_{10} - 110_{10} = 40_{10} = 50_8 = 28_{16} = 101000_2
2. `-13.0 * -6 % 13 / -12 => -0.0 -` Dividing a floating point zero value by a negative number results in -0.0.
3. The error is caused by the attempt to subtract 2 from the string, `"100F"`. Subtraction is not an operation defined for strings.
4. `"e","ce","science"` are all suffixes of `"computer science". "rscience"` and `"er"` are not.
5. Since both p and q are true, all parts of this statement are false due to the NOTs, therefore the entire expression is false.
6. The square root of 16 is 4.0, since the Math.sqrt function always returns a double.
7. Since 6 - 5.2 equals 0.8, and j does not store fractional parts of numbers, the 0.8 is truncated, leaving the value zero to be assigned to j. There is no error due to automatic casting that occurs with shortcut operations like this.
8. The sequence of x values is: 0,(+2 for "an")=2,(+1 for "eye")=3, (*2 and +1 for "for")=7, (+2 for "an")=9, (+1 for "eye")=10, (-1 for "a")=9, (+1 for "tooth")=10, (*2 and +1 for "for")=21, (-1 for "a")=20, and (+1 for "tooth")=21
9. The sequence of output values for this code is: 50 25 75 37 111 55 165 82 246 123 369 184. The last value of x is 552, which causes the loop process to end. The value 114 is not output in this code.
10. The value in position 3 of the first list is -3, and in position 2 of the second list is 52. It appears the lists switch contents, but in reality both lists end up referencing the second list, which contains -100 in position 3, and 52 in position 2.
11. The first statement does not compile - nextBool is not a valid statement. The second statement works, and so does the third one. The fourth statement compiles, but throws a runtime (NoSuchElementException) error when it reaches the nextDouble statement, since the previous nextLine statement gobbled up the rest of the line.
12. The sign of the result of any mod operation is always the sign of the first operand, regardless of the sign of the second operand, therefore, the only value in this list that results in a mod -3 value of -1 is the value -4.
13. The sequence of the expression evaluation is as follows: a becomes 11 and is added to b (10), which becomes 11 after the addition (sum of 21). The values of a and b are now both 11, but when compared in the second part of the expression, b changes to 12 first, making the comparison false.
14. The char data type ranges in integer values from 0 to 65536, and wraps back around to zero when it reaches the maximum value. Therefore the loop counter c reaches 65536.
15. The remove(new Integer(5)) method call removes the object whose value is 5, resulting in the contents of the array being: `9 3 7 2 4 1 6.`
16. Below is the truth table to evaluate this expression. It also simplifies quite nicely using the "disappearing opposite" rule, leaving AB+C, which matches the truth table results.

A	B	C	$A*B$	$\overline{A*B}$	$\overline{A*B}$	$A*B+\overline{A*B}$
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1

17. `10 - 18 % 7.0 ==> 10 - 4.0 ==> 6.0`
18. This code simply counts how many positive and negative whole numbers are in the list.
19. Use the two's complement conversion process to find the positive value, then just make it negative. 10011001 converts back to 01100111, which is the value 103, hence the original bit string is -103.
20. The cosine of a 60 degree angle is roughly 0.9 (0.866 to be more precise...opposite over hypotenuse). In a 30-60-90 triangle with short side of 1, hypotenuse of 2, and middle side of root 3, the calculation would be root 3 divided by 2. By process of elimination, choices A (too small), B(values are not identical), and D(too big) do not work, therefore, C is the only choice left.
21. This is the quick sort, which sorts a list of numbers in ascending order. The values 1 and 9 indicate the range of the list to sort, which in this situation excludes the first and last element of the list from being sorted.
22. The two parameter values indicate the exact low and high positions of the range of the list to be sorted. Since there are 11 elements in this list, positions 0 and 10 are the boundary positions of the entire list.
23. Lines 7 and 8 represent the comparisons to the pivot in the quick sorting process, which need to be reversed in order to reverse the sorting process to descending order.
24. The average case running time for this quick sort is O(N log N), which occurs when the list is random order.
25. The sequence of values in the process is: 0 50 0, 7 50 41, 14 50 32, 21 50 23, 28 50 14, 35 50 5, 42 50 -4, 49 50 41, 56 50 32, 63 50 23, 70 50 14, 77 50 5, 84 50 -4, 91 50 41, 98 50 32, 105 50 23, 112 50 14, 119 50 5, and **126 50 -4**, which is the set of values that is output.
26. This triple right shift operation is the equivalent of dividing the value by 8, or 2 to the power of 3.
27. This is a call to a binary search method. After the list is sorted, the ten elements of the arrays are: -7 -5 -3 0 1 3 4 5 8 9. The first mid position is the midpoint between the first and last positions: (0+9)/2, which is 4, the first output value. Since the target value 5 is greater than the value 1 in position 4, the search goes to the right of position 4, and finds the midpoint between positions 5 and 9, which is position 7, the second output.
28. `"aaabaab"`matches until the `"b"`, but fails after that since there is no subsequent `"a"` pattern provided.
29. There are four temperature values in this list (32, 53, 67, and 45) that fall below 70, which either output `"cool"` or `"cold"`.
30. In this inheritance situation, the correct call to the parent class, which requires a boolean parameter, is `super(b).`
31. Since "moll" is an invertebrate (`back` is `false`), and is a result of the case value 3 in the second switch statement of the Class constructor, the parameters required are (`false,3`).

32. The value `true` occurs three times when the mod value of x is 2 (when x equals 2, 7, and 12) , constructing and outputting the "`amph`" object three times, the mode (most often occurring value) of the output.
33. A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. The expression for the number of nodes in a full tree with N levels is 2^N-1, therefore the 40 nodes in a complete tree falls somewhere between a full tree with 5 levels (2^5-1=31 nodes) and a full tree with 6 levels (63 nodes), thus there are 6 levels in a complete tree with 40 nodes.
34. The decimal values 12, 21, 42, and 50 correspond to the hex values C, 15, 2A, and 32.
35. The list sequence is: 3, 3 9, 3 9 6, 3 9 6 7, ~~3~~ 9 6 7, ~~3 9~~ 6 7, ~~3 9~~ 6 7 5, ~~3 9 6~~ 7 5, ~~3 9 6~~ 7 5 2, ~~3 9 6 7~~ 5 2, with the value 5 at the front of the list at the end.
36. At first, the 'a' and 'b' map to "narf" and "goram, but are remapped to "slup" and "nox", thus resulting in the output, "`nox mork slup snuff`"
37. There are 16 paths of length 2, which are between these pairs of starting and ending nodes: AC, AE, BA, BC, CB, CC, CE, DB, DF, two for the pair EB (ECB and EAB), ED, EF, FB, FE, and FF. Either visually count carefully all the paths of length 2, or use matrix multiplication with M1 (the adjacency matrix for this graph) times M1 to get M2 (the matrix with paths of length 2). See below.

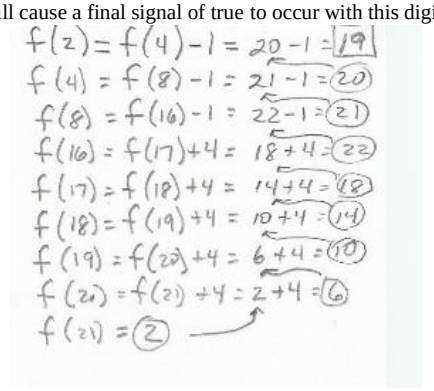
M1 X M1 = M2

	A	B	C	D	E	F
A	0	1	0	1	0	0
B	0	0	0	0	1	0
C	0	1	0	0	0	1
D	0	0	1	0	0	0
E	1	0	1	0	0	0
F	0	1	1	0	0	0

	A	B	C	D	E	F
A	0	1	0	1	0	0
B	0	0	0	0	1	0
C	0	1	0	0	0	1
D	0	0	1	0	0	0
E	1	0	1	0	0	0
F	0	1	1	0	0	0

	A	B	C	D	E	F	Totals
A	0	0	1	1	0	0	=2
B	1	0	1	0	0	0	=2
C	0	1	1	0	1	0	=3
D	0	1	0	0	0	1	=2
E	0	2	0	1	0	1	=4
F	0	1	0	0	1	1	=3
							=16

38. Three ordered triples (1,0,1), (1,1,0) and (1,1,1) will cause a final signal of true to occur with this digital electronics diagram.
39. Here is the trace for this recursive function call.



40. The equivalent infix expression for this prefix expression is **`W - O * R / L + D`**, and then to postfix: **`W O R * L / - D +`**