Computer Science Contest #1415-10 Key

January 17, 2015

1) B

2) E

3) C

4) A

5) B

6) A

7) C

8) B

9) C

10) A

11) D

12) B

13) B

14) A

15) D

16) B

17) D

18) C

19) E

20) C

21) D

22) C

23) E

24) D

25) E

26) C

27) D

28) C

29) E

30) B

31) C

32) E

33) B

34) B

35) E

36) D

37) E

38) B

39) 12

40) A

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:

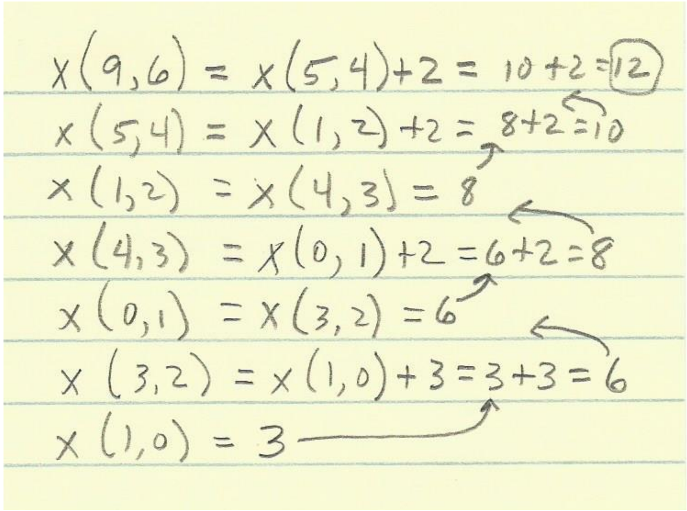
1. B $142_{10} - 86_{16} = 142_{10} - 134_{10} = 8_{10} = 10_8 = 8_{16} = 1000_2$
2. E 19 + 3 % 5 – 7 = 19 + 3 – 7 = 22 – 7 = 15
3. C The "%5.2f" field width is not sufficient to hold this number, therefore the compiler takes an extra space, outputting 130.56.
4. A All instances of the letter "I", followed by any character, are replaced by "OZ", resulting in `WOZTEROZCOMOZG`
5. B Since p and q are opposite values, the XOR result is true, and the NOT of that is false.
6. A The fourth root of 81 is 3.0.
7. C 3.16 * 10 = 31.6, which truncates to 31 in the autocast provided by *=
8. B -15 mod -3 is zero, therefore the output is the result of -15 divided by -3, which is 5.
9. C The value sequence of x is -100, -97, -94, -91, …-4, -1, 2.
10. A The original array is referenced by both list1 and list2, but list3 received a deep copy (identical list, but separate memory) of list1 using the clone method, therefore the change to list2 also affected list1, but not list3.
11. D The next command grabs the "3", the first nextLine takes the leftover whitespace as an empty String, and the final nextLine takes the "Cat".
12. B The sequence for a and b is: 1 2, 3 4, 7 8, 15 16, 31 32, 63 64, 127 -128. When 64 is doubled, equaling 128, it wraps around to -128 since 127 is the maximum value in the byte range.
13. B The *`expr++ expr--`* (postfix) operators are on line 1 of the chart, followed by `==` `!=` on line 7, and `&&` on line 11.
14. A A float value takes 32 bits to store, uses a maximum of 23 bits for decimal precision, which effectively allows for up to 7 decimal places of precision before loss of accuracy begins to occur.
15. D The addAll method places all aList2 elements into aList1, starting at position 4, which increases the size of aList1 to 15 elements.
16. B This is the merge sort, which sorts a list of numbers in descending order since the comparison operator in the if statement is >. The output is simply the choice that shows all of the numbers in order from greatest to least.
17. D The comparison operator in `<statement 4>` needs to be reversed to be `<if (list[i] < list[j])>` in order to sort the list in ascending order.
18. C This is an example of the merge sort.
19. E The running time for a merge sort in any case is O(N log N).
20. C To convert back from two's complement, use the same process to find the positive value, then just make it negative. 10000001 converts back to 01111111, which is the value 127, hence the original bit string is -127.
21. D 97 - 32 + 3.5 is equal to 68.5, which when cast as a char truncates the .5 and produces the character 'D'.
22. C `public static final int` creates an integer (int) constant value (final) accessible everywhere (public) as a member of the Mascot class (static).
23. E Either **`Map <School,Mascot> m = new HashMap<School,Mascot>();`** `or` **`Map <School,Mascot> m = new TreeMap<School,Mascot>();`** will correctly accomplish this task.
24. D When a mapping is made, the **entire object** maps to another **entire object**, using the put method receiving two parameters in the list.
25. E Both A and C work correctly.
26. C The split pattern `"\\D+"` means split on one or more (+) of any non-digit characters, i.e., letters, spaces, symbols other than 0-9.
27. D Both segments I and II will produce the output shown.
28. C Comparable is an interface and not a subclass of Object.
29. E Using the clone method produces a deep copy, therefore list[4] is a duplicate of list[0], but a completely separate array, which means the change to list[1] does not affect list[4]. However, the list[3]=list[0] is a shallow copy, which means both arrays are referencing the same array, and the change made to list[3] also affects list[0].
30. B The sequence of a and n values for this method call is: 0 1, 1 2, 3 3, 6 4, 10 5, 15 6, 21 7, 29 8, and 38 9.
31. C The sequence of a and n values for this method call is: 6 2, 8 3, 11 4, 15 5, 20 6, 26 7, 34 8, and 43 9.
32. E The stack contents sequence is as follows, with the end of the list being the top of the stack: [3], [3, 7], [3, 7, 6], [3, 7, 6, 1], [3, 7, 6], [3, 7, 6, 4], [3, 7, 6, 4, 5], [3, 7, 6, 4], [3, 7, 6, 4, 7], [3, 7, 6, 4], [3, 7, 6], [3, 7, 6, 2], [3, 7, 6]. The third value popped was 7, and 6 is at the top of the stack, waiting to be popped next.
33. B

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

The truth table above shows one false result in the final column. $\overline{B}+A*B$ also simplifies to $\overline{B}+A$ using the "disappearing opposite" rule.

34. B The first rule is that the operands stay in the same order, T R I T O N. Then insert the operators as shown. The ^ goes between the T and R, and then the first minus goes between the I and T in parentheses, * between the O and N, second minus between the T^R and (I-T), and finally the + between the (I-T) and O*N to complete the expression.
35. E The value 53 left shifted 3 times is the equivalent of 53*2*2*2, which in sequence equals 53*8, or 424.
36. D There is only one path of length two: AAC. The three paths of length 4 are: AAAAC, AACAC, and ACAAC.
37. E The four random values generated are 0, 1, 2, and 3, which when added to 45 are 45, 46, 47, and 48. 49 is not possible.
38. B The signals A and B go into an OR gate, which feeds into an AND gate that also receives a NAND signal from A and C. The AND result goes into an OR gate which receives another AND signal, with B as one feed and XNOR from C and D as the other.

39. 12 Below is a complete tracing of this recursive function call shown above.



$x(9,6) = x(5,4)+2 = 10+2 = \boxed{12}$

$x(5,4) = x(1,2)+2 = 8+2 = 10$

$x(1,2) = x(4,3) = 8$

$x(4,3) = x(0,1)+2 = 6+2 = 8$

$x(0,1) = x(3,2) = 6$

$x(3,2) = x(1,0)+3 = 3+3 = 6$

$x(1,0) = 3$

40. A

Explanation

$A * (A + \overline{B} * C) + \overline{B} * (B + A * \overline{C})$ ==> $A + \overline{B} * (B + A * \overline{C})$ - A absorbs the entire expression in parentheses (Rule 11)

$A + \overline{B} * (B + A * \overline{C})$ ==> $A + \overline{B} * B + A * \overline{B} * \overline{C}$ - NOT B distributes across the expression in parentheses (Rule 7)

$A + \overline{B} * B + A * \overline{B} * \overline{C}$ ==> $A + A * \overline{B} * \overline{C}$ - NOT B and B disappears (Rule 17)

$A + A * \overline{B} * \overline{C}$ ==> A - A absorbs entire second term (Rule 12)