Computer Science Contest #1718-13 Key

February 17, 2018

1) D

2) E

3) B

4) E

5) E

6) B

7) B

8) A

9) B

10) E

11) B

12) E

13) C

14) D

15) D

16) D

17) C

18) E

19) B

20) D

21) A

22) B

23) B

24) D

25) A

26) A

27) E

28) A

29) B

30) B

31) D

32) B

33) B

34) C

35) C

36) C

37) D

38) B

39) -86

40) (1,1,0)

**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. 3016 = 000000110000

+34916 = 001101001001

= 001101111001 = 1+8+16+32+64+256+512= 889

2. x/2 is an int divided by an int so it will return an int.

3. \\ to print \, - means left alignment.

4. k must start after the T, so you must add 1. p should not include the T, so you do not add anything to it as substring in exclusive on the right hand side.

5. DeMorgan’s Law describes the distribution of NOT across a quantity. We are used to seeing it as !(A&&B), but you must realize that A could be a condition with a NOT in it.

6. [-5,46] means -5..-1,0,1..46. So there are 52 unique values to find.

7. x was printed out, not sum. Always be aware of what is being printed.

8&9. Since there is no break on any of the cases, then whichever case is chosen, all cases after that case is also done. That is Z is appended 4 times and that is why X will be the most appended letter.

10. The array is not big enough so that when list[9] is called, it crashes.

11. It is entirely acceptable to have 2 Scanners linked to the same file.

12. This is a summation from the middle to the front.

13. Here is a great website for precedence in java

<https://introcs.cs.princeton.edu/java/11precedence/>

14. A double can store all of the integer data types.

15. the first String goes in the ArrayList in order, the second String goes in backwards (count never changes). Then remove will shrink the ArrayList by one, and you are still skipping by 2.

16&17. I like the Comparable, it allows me to resort Strings based on any way I want. In this case, I am comparing the back half of the String with the front half appended to it to another String, but only if they are the same length and not one letter long, all to the tune of an old Weird Al Yankovic song.

18. When you remove an element, the ArrayList shrinks and it is possible to go out of bounds from that. The test data does this.

19. This is a standard money conversion algorithm.

20. UF does not get printed out because the while jumps out before it can print.

21. d = 150 – 100 = 50 k = 50/100 = .5 k > .5 if false

k >= .25 is true 150 \* .15 = 15+7.5 = 22.5

50 \* .03 = .5+.5+.5 = 1.5 22.5+1.5=24.0

22. d = 430 – 300 = 130 k = 130/300 = 13/30 which is between 25% and 50%

430 \* .15 = 43+21.5 = 64.5 130\*.3 = 1.3+1.3+1.3 = 3.9

64.5+3.9 = 68.4

23. This is a B object so it will only printout 4 characters

24. This is a P object so it will printout 8 characters

25&26 This is definitely a time suck question. However question 25 can be done with one pass and no incident. Question 26 takes time. However, please note that list[0] and list[3] are both B classes. Further, you can anticipate what parts of the question affect list[3],

num=0 1 2 3 4 5 So, list[0] and list[3] an num = 2

i = 0 0 0 1 1 2 list[1] and list[3] at num = 4 🡪 this will be effected

j = 1 2 3 2 3 3 by list[0] and list[1] at num = 0, but that was Q25.

list[2] and list[3) at num = 5, but by this time, you

have a good guess at what the answer is.

27. m(-27) = -2 + m(-23) = -4 + m(-19) = -6 + m(-15) = -8 + m(-11) =

-10 + m(-7) = -10 + -14 = -24

28. m(22) same logic gets to 12 + 2\*4 = 20.

29. 7ABC16: 7 is 0111, A is 1010, B is 1011, C is 1100. ABC simple as 123.

30. the binary search tree will have 21 as its root and then must place ever value to the left or right based on its value compared to the parent node.

31. when printing out the entire map, it goes key = value.

32. the iterator is another way of going through a Set.

33. offer means to add, poll means to remove, peek means to get

34. There is only 32 bits available to an int, and the first bit is a negative, so when you move every bit over be 6 you get

1111 1111 1111 1111 1111 1111 1100 0000 which is going to be a negative number. To find this negative number you have to do 2s compliment.

0000 0000 0000 0000 0000 0000 0011 1111 and add 1 => 100 0000 = 64 but it will be a -64.

35. (e|l|o) means e or l or o.

36. !(!B(!(AB)+!AC))

B + !(!(AB)+!AC)

B + AB \* !(!AC)

B + AB \* (A + !C)

B + AAB + AB!C

B(1 + A + A!C)

B

37. RSHIFT-4(10010111 OR (LCIRC-3 00100111 AND LSHIFT-2 01010011))

RSHIFT-4(10010111 OR (00111001 AND 01001100))

RSHIFT-4(10010111 OR 00001000)

RSHIFT-4(10011111)

00001001

38. //A-BC\*D+EF

//A(B-C)\*D(E+F)

/(A/(B-C))(D\*(E+F))

(A/(B-C))/(D\*(E+F))

A/(B-C)/(D\*(E+F))

39. 10101010

01010101 swap bits

01010110 add one

2+4+16+64 = 86 which means it is -86

40.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | !A||B | A&&!C | !((!A||B)&&( A&&!C)) |
| T | T | T | T | F | T |
| T | T | F | T | T | F |
| T | F | T | F | F | T |
| T | F | F | F | T | T |
| F | T | T | T | F | T |
| F | T | F | T | F | T |
| F | F | T | T | F | T |
| F | F | F | T | F | T |