

Computer Science Contest #1415-12 Key

February 07, 2015

- |       |                   |
|-------|-------------------|
| 1) A  | 21) C             |
| 2) D  | 22) C             |
| 3) E  | 23) A             |
| 4) B  | 24) C             |
| 5) B  | 25) D             |
| 6) D  | 26) A             |
| 7) C  | 27) C             |
| 8) C  | 28) E             |
| 9) B  | 29) C             |
| 10) E | 30) B             |
| ■     | ■                 |
| 11) D | 31) D             |
| 12) A | 32) B             |
| 13) B | 33) B             |
| 14) B | 34) D             |
| 15) A | 35) C             |
| 16) B | 36) D             |
| 17) A | 37) A             |
| 18) E | 38) B             |
| 19) E | 39) $\bar{A} * B$ |
| 20) C | 40) 9             |
| ■     |                   |

**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:

- $4F_{16} + 111_8 = 79_{10} + 73_{10} = 152_{10} = 230_8 = 98_{16} = 10011000_2$
- $16 \% 9 - 4 * 0.2 = 7 - 4 * 0.2 = 7 - 0.8 = 6.2$
- Study them carefully...all of the choices will produce the given output.
- The first instance of "skt" at position 9 is replaced with "oed", producing "jeuskdjfhoeduryehdjskt"
- Since q is false, NOT q is true, which makes the entire expression true. This OR expression results in true, regardless of the result of the AND portion that is evaluated first.
- The nextAfter(double start, double direction) method returns the floating-point number adjacent to the first argument in the direction of the second argument, which means any of the choices shown between 47.00000000000004 to 47.00000000000009 are valid possibilities. The only choice that is not within that range is 47.00000000000035.
- Since the char value of '0' is 48, the resulting value for j is  $5 * 48$ , which is 240.
- Since  $15\%4$  is the value 3 and  $15/6$  is 2, the if statement is true, thus outputting "win".
- Since  $x+=4$  is evaluated before it is output, the first output is the value 7, and the final output is 23.
- The ++i increment step needs to be i++ for this code to work properly. As it is now, it causes a runtime "index out of bounds" error since it tries to access position 5 in list2, which does not exist.
- Since there are an odd number of values in the input string, the final iteration of this loop will attempt to access a data item that does not exist, thus causing a runtime error (NoSuchElementException).
- The sequence of output values for j and k are the choices given, from B to E. The initial values of 5 and 4.5 change before the first output.
- Since the additive operator (+, line 4 on the Java Operator Precedence chart) has precedence over the shift operators (<<,>>, line 5),  $2+2$  occurs first, resulting in  $3<<4>>3$ , then  $48>>3$ , then 6.  $3<<4$  is equivalent to  $3 * 16 (2^4)$ , and  $48>>3$  is equivalent to  $48 / 8 (2^3)$ .
- This is simply a true or false situation, and using the boolean data type is best for this purpose.
- The remove(5) method call removes the element in position 5, NOT the element whose value is 5, resulting in the contents of the array being: 9 3 5 7 2 1 6.
- The truth table below shows one true result in the final column.

$$\overline{B} \quad \overline{A} \quad \overline{B} \quad \overline{A+B} \quad \overline{\overline{A+B}}$$

0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

$\overline{\overline{A+B}}$  simplifies to  $A * B$  using DeMorgan's law and the Double Negative Rule, showing (1,1) as the only true result.

- Since the integer value of the character '9' is 57, the value of y is 9, therefore  $57.0 - 57 + 9$  is equal to 9.0.
  - The output is 3 2 2 6, representing the fact that this matrix has 3 rows, 2 elements in the row 2 (the last one), and 2.6 in row 1, col 1.
  - Using the two's complement short-cut conversion process, 10010001 converts back to 01101111, which is the base ten value 111 (one hundred eleven), hence the original bit string is -111.
  - As you can see by the chart below, there are 5 scenarios that are rated NlogN.
- | Algorithm Scenarios/Big O Time Complexity | Best  | Avg   | Worst |
|-------------------------------------------|-------|-------|-------|
| Quick Sort                                | NlogN | NlogN | N^2   |
| Merge Sort                                | NlogN | NlogN | NlogN |
| Insertion Sort                            | N     | N^2   | N^2   |
| Selection Sort                            | N^2   | N^2   | N^2   |
- Since the Insertion sort only sorts when necessary, a list in already sorted order will only take N comparisons to process using this sort.
  - The sequence of values shown is: 0 10 20 30 40 50 60 70 80 90
  - <statement1> is searching for the value 6 in the range of the sorted list (0,1,2,3,4,5,6,7,8,9) from positions 3 up to but not including 5. Not finding it in that range, the binary search reports it should be in position 5 by adding 1 to 5 and making it negative, hence -6 as the result.
  - <statement2> is searching for the value 3 in the range of the sorted list (0,1,2,3,4,5,6,7,8,9) from positions 2 up to but not including 9. The value is found in position 3.
  - The variable a is a formal parameter, and only exists during the execution of mystA5. The variable b is an actual parameter, passed by value to mystA5, NOT by reference, and exists before, during, and after the call to mystA5.
  - Despite the change to the variable a prior to the return statement in the method execution, since the variable b is passed by value, there is no change to b in the client code.
  - The split pattern "[a-z&&[^bc]]" means split on any lowercase letter, except for b or c. Each split resulting in an empty string is designated in the output loop by a single dash.
  - The implementation of an interface method must be public, with zero or more valid statements inside {}, and no semicolon after the }.
  - Same explanation as the previous question.
  - Since the two methods belong to the object of the class, the object must be stated first, followed by a period, and then the method call.
  - The sums after each pop are as follows: Stack = 7, 10, 16, 21, and 25; Queue = 3, 8, 15, 21, and 24. After the fourth pop, the sums are equal.
  - In a Priority Queue, the elements are stored in a min heap, as is shown in the second output. As each element is removed, the min heap is adjusted (re-heapified) so that the next item to be removed is in natural order, thus the third output is in alphabetical order.
  - The first statement with nested instantiations of ListNode objects is valid, with each nested node linked to the prior one. The separate instantiation in the second statement adds the new node to the front, linking the existing list to it.

34. The 15 possible values generated by this code range from 6 through 20.
35. In this schematic, the A signal first goes into the OR gate (the arrow) along with the NOT B signal, which then feeds into the AND gate (the bullet shape) along with the NOT A signal.
36. Although there is no guarantee of order in the iteration over a HashSet, the only issues in this problem are that **null** CAN be added to a HashSet, and since it is a set, there is only one occurrence of each value - no duplicates.
37. The hash set adds and retrieves any item in constant time, which is O(1).
38. The value 3 is found in position 2 of the array.
39. Rationale:

$$(A \oplus B) * (\bar{A} + B) \implies (\bar{A} * B + A * \bar{B}) * (\bar{A} + B) \text{ - Definition of XOR}$$

$$(\bar{A} * B + A * \bar{B}) * (\bar{A} + B) \implies \bar{A} * B * \bar{A} + \bar{A} * B * B + A * \bar{B} * \bar{A} + A * \bar{B} * B \text{ - FOIL}$$

$$\bar{A} * B * \bar{A} + \bar{A} * B * B + A * \bar{B} * \bar{A} + A * \bar{B} * B \implies \bar{A} * B + \bar{A} * B + \bar{B} * 0 + A * 0 \text{ - Rules 2 and 17}$$

$$\bar{A} * B + \bar{A} * B + \bar{B} * 0 + A * 0 \implies \bar{A} * B \text{ Rules 2 and 10}$$

40. To count the paths of length one in any directed graph, such as this one, simply count the arrowheads. In this graph, there are nine paths of length 1.