

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2022 INVITATIONAL A

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

1) <u> B </u>	11) <u> C </u>	21) <u> B </u>	31) <u> A </u>
2) <u> D </u>	12) <u> B </u>	22) <u> B </u>	32) <u> A </u>
3) <u> C </u>	13) <u> B </u>	23) <u> B </u>	33) <u> C </u>
4) <u> D </u>	14) <u> C </u>	24) <u> C </u>	34) <u> C </u>
5) <u> B </u>	15) <u> C </u>	25) <u> C </u>	35) <u> D </u>
6) <u> D </u>	16) <u> D </u>	26) <u> B </u>	36) <u> E </u>
7) <u> B </u>	17) <u> E </u>	27) <u> B </u>	37) <u> D </u>
8) <u> B </u>	18) <u> A </u>	28) <u> E </u>	38) <u> C </u>
9) <u> B </u>	19) <u> A </u>	29) <u> C </u>	*39) <u> 2 </u>
10) <u> A </u>	20) <u> B </u>	30) <u> B </u>	*40) <u>10000000</u>

** See "Explanation" section below for alternate, acceptable answers.*

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	$1 + 8 + 32 + 256 = 297$
2.	Order of operations and integer division, $(9*8) + ((7 \% 6) / 5) = (72) + (0) = 72$
3.	Round to 2 decimals and print at least 5 characters, zero not space -> 02.35
4.	The last o to appear in str is at index 7
5.	Chained xor is true if there are an odd number of trues, false otherwise
6.	2.45 rounds down to 2 (Math.round returns a long)
7.	$(6 * 1.5) + (10 / 1.25) = (9.0) + (8.0) = 17.0$
8.	There is no break after the first case, so the second case will execute as well
9.	3 loop iterations, each adding a star to str and then printing. 1 star + 2 star + 3 star = 6 stars total
10.	Initial list: [6,5,4,3,2,1] After first move: [6,5,4,3,1,1] After second move: [6,5,1,3,1,1] (be careful to reference the altered array instead of the original)
11.	nextLine() will pick up only the carriage return after calling nextInt(), so no numbers are skipped
12.	A H 1 0 2 1 3 3 4 6 5 10 6 15 After all loop iterations, H = 15
13.	Order of operations, additions and ++b will come before shifts a will be incremented after the operation is completed $4 + 3 >> 1 = 7 >> 1 = 3$
14.	Floats are 32 bits
15.	ArrayList.remove(x) removes index X, not value X when given a primitive int
16.	Only one way to make A+B false, and 3 ways to make C+D true. The same math applies to the opposite combination. $1*3 + 3*1 = 6$
17.	$x -= y$ makes x zero then they swap so $x=y$ leaves x unchanged with y zero so condition is false and nothing changes. This leaves us in an infinite loop
18.	The code is finding GCD using Eulers method. GCD(108, 24) is 12. Tracing will find that this takes 5 iterations to converge
19.	Inner loop will execute 4 times per outer loop, outer loop iterates 10 times. $10*4 = 40$. Take note of the loop pointer doubling at each iteration instead of increasing by one
20.	Inner loop will still execute 4 times per outer loop, $4*16 = 64$
21.	Due to doubling each time, inner loop is logarithmic with respect to n. outer loop is linear, so overall is $N\log N$
22.	The random*2 will generate a real number in the range [0, 2), plus 10 will be in [10, 12). Truncating will result in only 10 or 11
23.	Addition comes before shifting, $4 << 1 = 8$
24.	Floor is the largest element less than or equal to the input
25.	Higher is the smallest element strictly greater than the input
26.	Assuming Math.random() is relatively fair and unbiased, on average each iteration will contribute .5 to the average. In the long run after 1000000 iterations, this will converge to a .5 average.
27.	Adding always sets the head to the new value, and removing and calling next returns the head element. Hence we are using a LIFO data structure. Take this into account and trace the code
28.	While loop removes all items, stack is empty with head null. If head is null, h.elem will throw a null pointer exception
29.	LIFO data structure is Stack
30.	The O class encapsulates the data of a node
31.	Min Value is -2^{32} , and max value is $2^{32} - 1$. Hence, their sum is -1

32.	<p>This simulates BFS on a 3x3 grid starting at (1,1) with an initial cost of 1, allowing up, down, left, right as directions.</p> <p>The entire grid will be:</p> <p>[3,2,3] [2,1,2] [3,2,3]</p>
33.	PriorityQueue uses a min heap internally, so sorting with PQ is a heapsort
34.	for the result to be false, C must be true. (A*B) must also evaluate to false, which there are 3 ways to do. hence the answer is 3
35.	<p>To convert to prefix from postfix, we look at the groups each operator is associated with, and move the operator to the front instead of the back. Order of letters must be preserved.</p> <p>The * applies to all 5 symbols The / applies to only D and E The – applies to A B and C The + applies to only A and B</p>
36.	first loop inserts 9 items, second loop attempts to remove 9 items but is increasing location by +2 while array is shrinking. the location exceeds list size and causes an exception.
37.	This marks an array index i true if i is a multiple of some smaller index encountered other than 1. Hence composite number will all be marked to true, and primes will be false
38.	$17 \& 1 = 1$, so we will evaluate the first half of the ternary operator. $17 \mid 14 = 10001 \mid 01110 = 11111 = 31$
39.	<p>Removing the edge 2-4 will disconnect (3,4) and (0,1,2)</p> <p>Removing the edge 3-4 will disconnect (0,1,2,3) and (4)</p>
40.	The last bit of a 2's complement number can be viewed as a negative 2^n . Hence the smallest value that can be made is only the last bit and no positive bits to increase it. Hence, 10000000.