WYSE - Academic Challenge
Computer Science Test (Sectional) - 2019

1. **PL/SEC** Here is a trace:

a mem	a curIndex	b mem	*c mem	*c curIndex	Comment
	0				
	0			0	
	0		77	1	Set mem, postfix incr. curIndex
4	1		77	1	Set mem, postfix incr. curIndex
4, 2	2		77	1	Set mem, postfix incr. curIndex
4, 2	2	99	77	1	push back val
4, 2	2	99, 17	77	1	push back val

DumpList a	DumpList tmp	Output	Comment
			Call DumpList(*c);
77	0		
77	1	77	Print and increment tmp
77	1	77	DumpList return
			Call DumpList(a);
4, 2	0		
4, 2	1	4	Print and increment tmp
4, 2	2	4, 2	Print and increment tmp
4, 2	2	4, 2	DumpList return
			Call DumpList(b);
99, 17	0		
99, 17	1	99	Print and increment tmp
99, 17	2	99, 17	Print and increment tmp
99, 17	2	99, 17	DumpList return

2. **PL/SEC** In this instance, `list<T>` is composed of pure virtual functions, and as such, cannot be directly instantiated. A class that provides implementations for all of the pure virtual functions, however, could be instantiated, such as `list<T> *someList = new CircularArray<int>();`. All classes have constructors.

3. **OOP/SEC** `DumpList` accepts an instance of an object which derives from the `list<T>` class. Additionally, whatever object is passed as the template parameter for `T` must also have overridden the insertion operators. This represents polymorphism because we can pass any derived object of the `list<T>` class, and any object for the `T` template parameter. Abstraction would involve hiding details of the implementation from a calling object. Encapsulation involves limiting access to private member variables through the use of getters and setters.

4. **OOP/SEC** Both `VectorList` and `CircularArray` inherit from `list<T>`. They are not directly related to one another. An Is-A relationship represents a situation in which one class inherits from another class. A Has-A relationship represents a situation in which one class is a property on another class.

5. **PL/SEC** Since `CircularArray<int>` is a derived class of `list<int>`, we can assign a pointer of the latter to an address of an instance of the former. We cannot directly assign an instance of one to another. Additionally, this only works in one direction; i.e., one cannot assign the address of an instance of a `list<int>` (if it were instantiatable) to a pointer of type `CircularArray<int>`, since the "Is-A" relationship is directional.

6. **PL/SEC** The function uses a linear search to find the maximum value in the list. Here is a trace of the code:

a	tmp	i	output	comment
99, 17				Call someFunc
99, 17	99			Set t
99, 17	99	1		increment i
99, 17	99	1		tmp $\not<$ a.Get(i)
99, 17	99	2		incr. i, exit for loop
99, 17	99	2		Return tmp
99, 17	99	2	99	Print result

7. **DSA/SEC** Since `CircularArray<T>` has a constant lookup time, we can concern ourselves solely with the performance of `someFunc`. Since the function loops from $i = 1 \to n$, we can say that the time complexity of the function is $O(n)$.

8. **GEN/SEC** Secure Shell (SSH) is a secure networking protocol used primarily for users to access remote servers. It is the secure successor to telnet, which provides insecure shell access. HTTP traffice is secured using TLS/SSL. There are various printing protocols, but an example is IPP (Internet Printing Protocol).

9. **DLNS/SEC** The circuit forms a NAND gate. When both inputs (A and B) are in the HIGH state, the output C is pulled LOW. However, if at least one of the input gates is LOW, then the output C will be HIGH.

10. **DLNS/SEC** We can add the two numbers in binary, then convert the result to decimal. We have: $00111011 + 11101111 = 100101010$. Next, we ignore any overflow bits, and we are left with the result in binary, which is 00101010. Since the most significant bit is 0, we know the value is positive, so we can simply convert to decimal using the normal binary-to-decimal conversion formula:

$$\begin{aligned} 00101010_2 &= 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 \\ &= 32 + 8 + 2 \\ &= 42 \end{aligned}$$

11. **GEN/SEC** HTTPS is used for secure web traffic over HTTP. SMTPS (Secure Simple Mail Transport Protocol) is used to securely send email between mail hosts. SSH (Secure SHell) is used to securely access remote terminals (common on Linux/Unix servers) as well as for securely copying and accessing files (common with applications such as git). LDAP (Lightweight Directory Access Protocol) is a protocol for accessing directory information systems, commonly used to manage users and devices in corporate networks, but is not inherently secure (though secure extensions to LDAP do exist).

12. **PL/SEC** The function `bubbleSort` sorts an array in ascending order. Here is a trace:

a	size	changed	i	tmp	comment
7, 4, 1, 9	4				Call bubbleSort
7, 4, 1, 9	4	F			
7, 4, 1, 9	4	F	1		
4, 7, 1, 9	4	T	1	4	a[i - 1] > a[i]
4, 7, 1, 9	4	T	2	4	
4, 1, 7, 9	4	T	2	1	a[i - 1] > a[i]
4, 1, 7, 9	4	T	3	1	
4, 1, 7, 9	4	T	3	1	do-while: changed = true
4, 1, 7, 9	4	F	3	1	
4, 1, 7, 9	4	F	1	1	
1, 4, 7, 9	4	T	1	1	a[i - 1] > a[i]
1, 4, 7, 9	4	T	2	1	
1, 4, 7, 9	4	T	3	1	
1, 4, 7, 9	4	T	3	1	do-while: changed = true
1, 4, 7, 9	4	F	3	1	
1, 4, 7, 9	4	F	1	1	
1, 4, 7, 9	4	F	2	1	
1, 4, 7, 9	4	F	3	1	
1, 4, 7, 9	4	F	3	1	do-while: changed = false, exit

13. **PL/SEC** Using the trace from the previous answer, the while loop conditional is checked three times.

14. **DSA/SEC** A stable sorting algorithm will maintain the original ordering of two equal values. This function will produce a correctly sorted array of values, regardless of length.

15. **GEN/SEC** TCP port numbers are transmitted in the TCP Header of a TCP Segment. Each header defines a source port and a destination port, each of which are allotted 16 bits. Therefore, the largest possible port number is 2^{16}.

16. **GEN/SEC** Network Address Translation provides a mechanism for translating IP addresses (most frequently IPv4, rarely IPv6) on network traffic as it passes between different networks. This is usually done on a network router, and a common example is a home router. When a device is on a home network, it usually has a non-routable IPv4 address (usually somewhere in the 192.168.0.0/16 block of addresses). If that device tries to send data outside of the home network, the packets are first routed to the home's router, which translates the non-routable address to that router's public IP address, before sending the packets on to the home's ISP. The router also reroutes incoming packets from the internet back to the original internal device.

17. **DLNS/SEC** A Karnaugh Map (or K-Map) is useful when simplifying a Boolean expression. This is beneficial when the Boolean expression is implemented with logic gates, as a minimal number of logic gates can be used. To get the minimal Sum Of Products from a K-Map, group all the True cells into the largest groups possible, where each group is rectangular (namely, its dimensions are nxm, where n and m are positive integers. Overlapping groups are permitted, and groups can wrap around the edges of the grid.

18. **DLNS/SEC** Converting Octal (base-8) numbers to decimal is similar to converting binary

to decimal, except with a different base. In this example:

$$\begin{aligned} 724_8 &= 7 \cdot 8^2 + 2 \cdot 8^1 + 4 \cdot 8^0 \\ &= 7 \cdot 64 + 2 \cdot 8 + 4 \cdot 1 \\ &= 448 + 16 + 4 \\ &= 468 \end{aligned}$$

19. **PL/SEC** Calling the `preferredMovingVerb` method on an instance of `Vehicle` simply prints the word "Drive\n". Since the `Airplane` class publicly inherits from `Vehicle`, and since it has a method of the same name, its `preferredMovingVerb` overloads that of the `Vehicle` class. In this case, it prints the word "fly\n".

20. **OOP/SEC** As previously mentioned, the `preferredMovingVerb` function overloads the function of the same name in the `Vehicle` class because `Airplane` publicly inherits from `Vehicle`.

21. **PL/SEC** The `Vehicle` class's operator overload for the Greater-Than operator returns true if the left-hand-side's wheel count is greater than the wheel count of the right-hand-side. In this case, `a` has a wheel count of 4 (since we did not explicitly pass in a wheel count when it was constructed, and the wheel count defaults to 4), and `b` has a wheel count of 3. This means that `a > b` evaluates to true, and the first `cout` statement is executed.

22. **PL/SEC** A default constructor is a constructor that accepts no parameters. If a class is defined without an explicit constructor, and implicit default constructor is provided. In this case, `Airplane` has a default constructor, but the `Vehicle` has a constructor which has a parameter with a default value. In C++ destructors cannot have parameters, so the concept of a default destructor doesn't really exist; all destructors have zero arguments. If an instance of `Vehicle` is instantiated with -1 for the `wheelCount`, in the constructor, the line of code `wheels = wheelCount >= 0 ? wheelCount : 0;` will set the `wheels` variable to zero.

23. **DSA/SEC** An Adjacency Matrix represents the connections between nodes of a graph. In the case of a graph with non-directional connections, the Adjacency Matrix is symmetrical along the diagonal, with zeros on the diagonal (a node is not considered to be connected to itself).

24. **DSA/SEC** A breadth-first search visits all the nodes connected to the start node. Nodes that are two connections away are visited next, and so forth. A depth-first search works by visiting nodes as far as it can until it reaches a dead end (i.e., a node with no un-visited neighbors), before backtracking the path until it finds another un-visited neighbor. In either case, the starting node would be "searched" first, not last.

25. **PL/SEC** The expression can be rewritten as follows to make it more clear: `4 << ((--a) - (a / b))`. With the prefix decrement, we can again rewrite and plug in known values: `(4 << (3 - (3 / 2))`. Next, we compute the division, then subtract that from 3, resulting in `4 << 2 = 16`. Note that `3 / 2 = 1`, since we are using integer arithmetic.

26. **PL/SEC** A ternary operator is an operator that accepts 3 arguments. In C++, there is only one ternary operator, `? :`, which is a conditional expression.

27. **PL/SEC** Here is a trace of the code:

function call	a	comment
fun1(11)	11	Call fun1
	11	a >> 2 != 0, return fun1(a / 2)
fun1(5)	5	Call fun1
	5	a >> 2 != 0, return fun1(a / 2)
fun1(2)	2	Call fun1
	2	a >> 2 == 0, return a

28. **PL/SEC** Using the previous trace, the function `fun1` is called three times.

29. **PL/SEC** The function is recursive because it calls itself. Recursive functions have base cases which are used to stop from continuing to recursively call themselves. In other words, without base cases, recursive functions would not end, until the computer terminates the program due to the stack and/or heap filling up.

30. **PL/SEC** An insertion operator is most commonly used with `cout`. The extraction operator is most commonly used with `cin`. The arrow operator is used when accessing a member of an instance that is pointed to by a pointer. The modulus operator returns the remainder of a division. The shift operator takes two integers, and shifts the bits of the left-hand-side by the number of bits specified by the right-hand-side.