WYSE – Academic Challenge
Computer Science Test Solutions (State) – 2018

1. Correct Answer: D
SOLUTION
Persistent storage is storage that would not lose the data stored on it after losing power. Hard disk drives and solid-state drives are both examples of storage medium which are used for their persistence. Volatile memory, on the other hand, is any storage medium that would lose the data stored on it if it lost power. Examples of volatile memory include CPU registers, cache memory, and RAM. Random Access Memory (RAM) generally has much faster access times than hard disk drives due to the fact that there are no moving parts. CPU registers generally contain small amounts of data, but are used inside of the CPU to provide very fast, temporary storage space. A typical computer will take advantage of several types of memory, which all have their strengths and weaknesses. This is commonly referred to as the Memory Hierarchy. Faster memory mediums, such as cache and RAM are usually more expensive per byte; however, slower storage mediums, such as hard disk drives, are usually much cheaper per byte, and are non-volatile.

2. Correct Answer: A
SOLUTION
A Class A network has a range of 0.0.0.0-127.255.255.255, with a netmask of 255.0.0.0. The netmask for a Class B network is 255.255.0.0, and the netmask for a Class C network is 255.255.255.0.

3. Correct Answer: A
SOLUTION
We can break the IP addresses into a string of 32 bits to determine which addresses fall within its network, given a subnet mask. Once we have the IP address and netmask converted to a bit string, we can calculate the range of IP addresses within that network. A “1” in the binary netmask string indicates the corresponding bit in the IP address cannot change for it to be considered part of the same network
10 = 0000 1010
206 = 1100 1110
0 = 0000 0000
30 = 0001 1110
We can rewrite the address and netmask as follows:

```
00001010 11001110 00000000 00011110    IP address
11111111 11111111 11111111 11110000    Netmask (/28)
```

Therefore, the allowable range of IP Addresses is:

```
00001010 11001110 00000000 00010000
00001010 11001110 00000000 00011111
```

Converting back to decimal:

```
10       206      0        16
```

```
10        206       0         31
```

Therefore, the range of IP addresses within the computer's network are 10.206.0.16-10.206.0.31. Communication to all other IP addresses would need to be routed to the computer's gateway router. Also note that the loopback address (127.0.0.1) would not be routed to the gateway, as it typically represents the loopback interface.

4. Correct Answer: C

SOLUTION

If a computer represents memory addresses with 4 bytes, it uses $4 \cdot 8 = 32$ bits. We can then calculate the total number of addressable memory locations using $2^{32} \approx 4.3 \times 10^9$.

5. Correct Answer: C

SOLUTION

The top logic gate represents an XOR, and the bottom represents an AND gate. Therefore, S = A XOR B, and C = A AND B

A	B	S = A XOR B	C = A AND B
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T

The circuit is implementing a binary half-adder circuit.

6. Correct Answer: E

SOLUTION

We can simplify the expression as follows:

```
!(x and (x or y)) or x
!x or !(x or y) or x        DeMorgan's Law
!x or x or !(x or y)        commutative property
true or !(x or y)           x or !x is always true
true
```

7. Correct Answer: A

SOLUTION

```
x or (!y and z) or (!x and z)
x or (!y and false) or (!x and false)      Substitute false
x or false or false                        x and false is
                                           always false
x                                          Simplify
```

8. Correct Answer: E
SOLUTION
To find the minimum Sum of Products expression, we would first need to make sure the columns and rows are ordered in gray code (which this problem already is). We begin by circling the largest group of true cells. The goal is to include all true cells in the minimum number of boxes.

	AB	AB'	A'B'	A'B
CD	F	F	F	**T**
CD'	**T**	F	**T**	**T**
C'D'	**T**	F	**T**	**T**
C'D	**T**	F	F	**T**

So, the minimum sum of products expression would be A'D' + BD' + BC' + A'B. Note that it is perfectly acceptable for a cell to be included in multiple groupings.

9. Correct Answer: B
SOLUTION
A tree is a special type of graph, in which there are no cycles. A binary search algorithm would be best applied to a balanced binary search tree. Examples of algorithms for searching a graph include a breadth-first search and a depth-first search algorithm.

10. Correct Answer: C
SOLUTION
When expressing the time complexity of an algorithm with Big-Oh, we only concern ourselves with the fastest growing part of the expression for large values, namely as $n \to \infty$. The equation can be broken up into the four components and then ranked:

$$2^n > n^2 > n > log(n)$$

So, the Big-Oh notation would be $O(2^n)$.

11. Correct Answer: D
SOLUTION
An adjacency matrix has a '1' if there is a connection between two corresponding nodes (i.e., if they are adjacent), and a '0' if there is no connection. A node isn't considered to be connected to itself unless a loop path is explicitly drawn on the graph.

12. Correct Answer: B
SOLUTION
When external to a class (and not in the context of a friend function), only public members are accessible.

13. Correct Answer: E
SOLUTION
Within a class, any public, protected, or private member of that class is accessible. Additionally, any protected or public members of a class are accessible to any class which inherits it. If a class inherits another class with the private modifier, then members from the superclass will not be accessible to class that inherit from it.

14. Correct Answer: D
SOLUTION
If class `B` was changed to inherit class `A` with the private modifier, then all members of class `A` are treated as if they were private in the context of any class that inherited from class `B`, or which had an instance of class `B`.

15. Correct Answer: D
SOLUTION
The code represents inheritance. Abstraction involves generalizing functionality. Encapsulation involves hiding the implementation details away from outside systems. Data hiding is similar to encapsulation in that it hides the data it cares about from the outside world. The Principle of Least Privilege refers to the idea that a piece of an application should only have access to the data it needs to do its job.

16. Correct Answer: C
SOLUTION
Note that `size_x = 3`, and `size_y = 5` throughout the code.

```
y   x   a   z                    comment
0   ?   ?   {?, ?, ?, ?, ?}
0   ?   0   {?, ?, ?, ?, ?}
0   0   0   {?, ?, ?, ?, ?}
0   0   1   {?, ?, ?, ?, ?}       c[x][y] = 1
0   1   1   {?, ?, ?, ?, ?}
0   1   3   {?, ?, ?, ?, ?}       c[x][y] = 2
0   2   3   {?, ?, ?, ?, ?}
0   2   6   {?, ?, ?, ?, ?}       c[x][y] = 3
0   3   6   {?, ?, ?, ?, ?}
0   ?   6   {2, ?, ?, ?, ?}       6/3 = 2
1   ?   0   {2, ?, ?, ?, ?}
1   0   0   {2, ?, ?, ?, ?}
1   0   2   {2, ?, ?, ?, ?}       c[x][y] = 2
```

```
1   1   2   {2, ?, ?, ?, ?}
1   1   5   {2, ?, ?, ?, ?}       c[x][y] = 3
1   2   5   {2, ?, ?, ?, ?}
1   2   9   {2, ?, ?, ?, ?}       c[x][y] = 4
1   3   9   {2, ?, ?, ?, ?}
1   ?   9   {2, 3, ?, ?, ?}       9/3 = 3
2   ?   0   {2, 3, ?, ?, ?}
2   0   0   {2, 3, ?, ?, ?}
2   0   3   {2, 3, ?, ?, ?}       c[x][y] = 3
2   1   3   {2, 3, ?, ?, ?}
2   1   7   {2, 3, ?, ?, ?}       c[x][y] = 4
2   2   7   {2, 3, ?, ?, ?}
2   2   12  {2, 3, ?, ?, ?}       c[x][y] = 5
2   3   12  {2, 3, ?, ?, ?}
2   ?   12  {2, 3, 4, ?, ?}       12/3 = 4
3   ?   0   {2, 3, 4, ?, ?}
3   0   0   {2, 3, 4, ?, ?}
3   0   4   {2, 3, 4, ?, ?}       c[x][y] = 4
3   1   4   {2, 3, 4, ?, ?}
3   1   9   {2, 3, 4, ?, ?}       c[x][y] = 5
3   2   9   {2, 3, 4, ?, ?}
3   2   15  {2, 3, 4, ?, ?}       c[x][y] = 6
3   3   15  {2, 3, 4, ?, ?}
3   ?   15  {2, 3, 4, 5, ?}       15/3 = 5
4   ?   0   {2, 3, 4, 5, ?}
4   0   0   {2, 3, 4, 5, ?}
4   0   5   {2, 3, 4, 5, ?}       c[x][y] = 5
4   1   5   {2, 3, 4, 5, ?}
4   1   11  {2, 3, 4, 5, ?}       c[x][y] = 6
4   2   11  {2, 3, 4, 5, ?}
4   2   18  {2, 3, 4, 5, ?}       c[x][y] = 7
4   3   18  {2, 3, 4, 5, ?}
4   ?   18  {2, 3, 4, 5, 6}       18/3 = 6
```

17. Correct Answer: E

SOLUTION

The time complexity can be expressed by looking at the `for` loops:

```
for(int y = 0; y < size_y; y++)          n
      for(int x = 0; x < size_x; x++)    n*n
            …
for(int i = 0; i < size_y; i++)          n
```

So, the time complexity can be expressed as $n^2 + n = O(n^2)$.

18. Correct Answer: D
SOLUTION
A function in C++ cannot return an array with a return type of `int[]`. Instead, it would have to be returned as a pointer to an array.

19. Correct Answer: B
SOLUTION
The caret (^) is used as an XOR (exclusive-or) in C++. Here is a truth table:

A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0

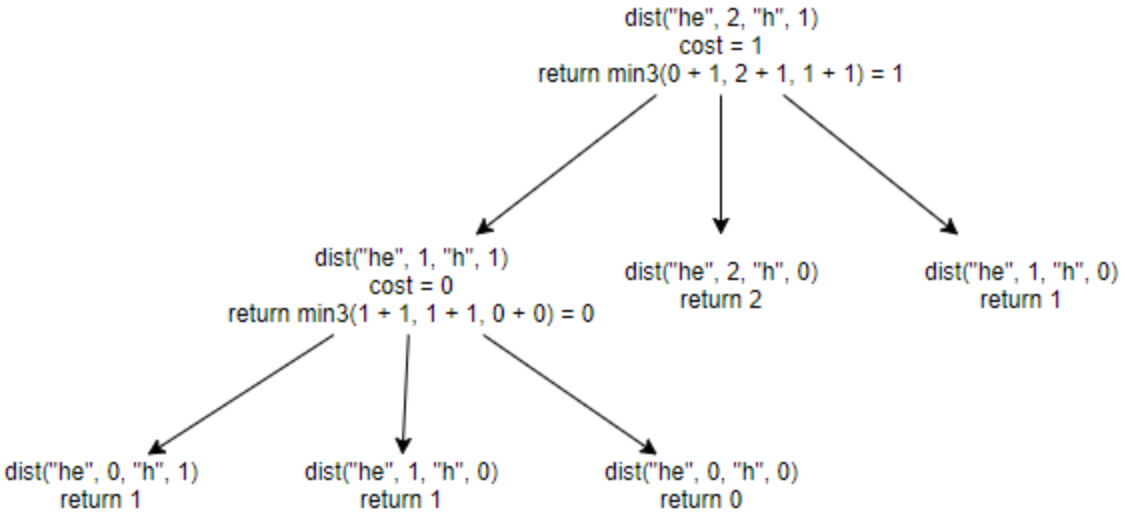
We can rewrite the two values as binary, then XOR them:

```
a = 0 0000 0100 ^ 1 0100 0000 = 1 0100 0100
b = 1 0100 0100 ^ 1 0100 0000 = 0 0000 0100
a = 1 0100 0100 ^ 0 0000 0100 = 1 0100 0000
```

The result is `a = 1 0100 0000 = 320`, and `b = 0 0000 0100 = 4`. Using the XOR operation three times on two variables as shown is a method of swapping the values of two variables without requiring a third variable as temporary storage.

20. Correct Answer: E
SOLUTION
Since XOR is a commutative operation, the order of the operands does not affect anything. Therefore, switching `b = a ^ b` to `b = b ^ a` will not affect the result.

21. Correct Answer: B
SOLUTION
The code computes the Levenshtein distance between two strings of text. Here is a trace of the function calls:



The final value returned will be 1.

22. Correct Answer: A
SOLUTION
The function is recursive since it calls itself. The base cases are used to keep the function from continually calling itself.

23. Correct Answer: D
SOLUTION
With the input of `("Go!", 3, "", 0)`, the code is easier to trace as there will be no recursive calls to distance.

```
*s        len_s     *t     len_t
"Go!"     3         ""     0
Return len_s = 3, since len_t is 0, which is a base-case.
```

24. Correct Answer: E
SOLUTION
Removing the base-case would not cause a compile-time error to be raised. Since there is still a base-case, the function would behave the same as before under certain circumstances, namely when `len_t <= len_s`, which would cause the function to hit the second base case anyways. However, if `len_s < len_t`, then the function would attempt to access an array element outside of the bounds of the s array. Under certain circumstances, this could lead to an incorrect result, or the program could result in a segmentation fault, depending on what memory is allocated to the program and what data is stored adjacent to the arrays.
If the line of code were removed, it would still sometimes return the correct result. As an example, consider the function call: `distance("Go!", 3, "", 0)` from the previous problem.

25. Correct Answer: B
SOLUTION
Declaring `const` on a parameter primarily does two things:

1. The variable cannot change within the context of that function, or any function it calls
2. Code that calls the function can operate under the assumption that those variables will not be mutated within that function call, which is made more relevant by the fact that pointers are being passed in.

26. Correct Answer: A
SOLUTION
Because each node in the element has a pointer to the next element in the list and a pointer to the previous element in the list, it is considered a doubly-linked list. This list could be used to implement a stack, queue, or any number of other data structures. However, in the context of the question, it is simply used to store a list of elements, then print data from each element to the screen.

27. Correct Answer: A
SOLUTION
Answer A is the only answer that correctly sets up all `prev` and `next` pointers to the correct nodes. Answer A points the second node's previous pointer to a `new` node, then uses that link to update the new node's `next` and `prev` pointers appropriately. Next, it updates `head`'s `next` pointer to point to the new node. Finally, it assigns a value to the `Age` property. Answers B and C are incorrect because they immediately assign `head`'s `next` pointer to the new node. By doing this, we lose a reference to the second node originally in the linked list. Answer D starts off by assigning the second node's previous pointer to the new node, but then updates `head`'s `next` pointer to the inserted node, before the inserted node's `next` pointer is set to point to the next node. As such, we again lose a reference to the node that has the `Age` of 30.

28. Correct Answer: C
SOLUTION
Answer A incorrectly accesses the `Age` property, as it would need to use "`sum += ptr->data.Age`" (which is the only difference between answers A and C). Answer B also incorrectly accesses the `Age` property on the data object. Additionally, it would not include the last element in the list when adding, because the conditional on the while-loop is incorrect. Answer C is correct, as it will add up all elements in the list (including the first and last elements), and uses the correct syntax for accessing the `Age` property. Answer D would result in an infinite loop, as it never reassigns `ptr` to the next node in the list. Answer E would not add the last element in the list.

29. Correct Answer: C
SOLUTION
The function fun computes the minimum value in an array. Here is a trace:

```
x                          a     i
{432, 88, 19, 77, 500}     432   ?
{432, 88, 19, 77, 500}     432   1
{432, 88, 19, 77, 500}     88    1
{432, 88, 19, 77, 500}     88    2
{432, 88, 19, 77, 500}     19    2
{432, 88, 19, 77, 500}     19    3
{432, 88, 19, 77, 500}     19    4
{432, 88, 19, 77, 500}     19    5
      Return 19
```

30. Correct Answer: E
SOLUTION
Similar to question 29, here is a trace:

```
x                          a     i
{0, 30, 22, 17, -2}        0     ?
{0, 30, 22, 17, -2}        0     1
{0, 30, 22, 17, -2}        0     2
{0, 30, 22, 17, -2}        0     3
{0, 30, 22, 17, -2}        0     4
{0, 30, 22, 17, -2}        -2    4
{0, 30, 22, 17, -2}        -2    5
      Return -2
```