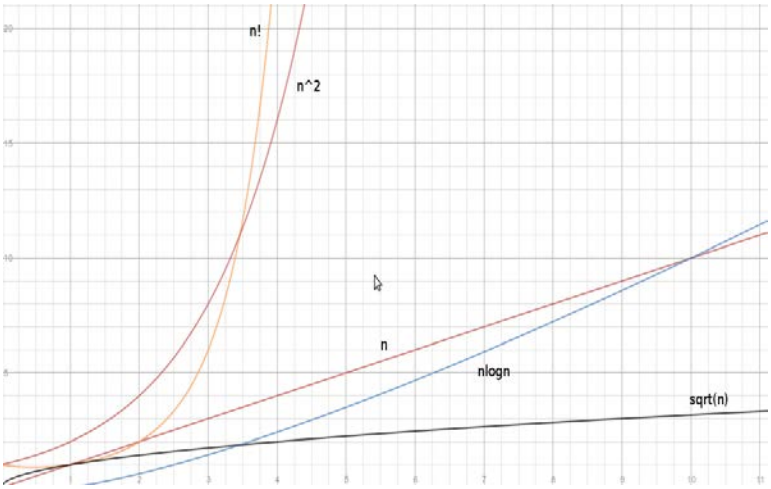
WYSE – Academic Challenge
Computer Science Test (State) – 2016

1. **Correct Answer: B**

The graph illustrates the growth of the functions involved. As n gets large, the factorial will grow faster than any of the other functions.



2. **Correct Answer: E**

The trace of the code follows:

```
node *head, *n;
head = new node;            // create new node that pointer points
to
head->data = 5;             // put 5 into that node
head->next = new node;      // create a new node that head->next
points to
head->next->data = 7;       // put 7 in the new node created
n = head->next;             // point n at head->next
                            // NOTE: there are only two nodes
n->data = 8;                // This overwrites the 7 with an 8
n->next = NULL;             // terminates the list
```

3. **Correct Answer: C**

a) will result in losing the first node that was in the previous list.
b) will lose the entire contents of the previous list.
c) works because it creates a new node, assigns its data value, points the end of this node to the previous list and renames the head pointer.
d) will lose the contents of the previous list with the exception of the first element. It then creates a new node point puts 11 into it and points the first element back to itself losing the second node.
e) puts the element to the front of the list, but does not point the head to the new front of the list.

4. **Correct Answer: D**

Hexadecimal numbers with a floating point portion just use the same power of 16 technique with the power decremented moving to the left. So the C would be in the 16^-1 spot. Some of the numbers can be a bit tricky to determine without a calculator, but since C is 12 decimal, 12/16 = 0.75 which is fairly easy to determine without one.

A7.C = $10*16^{*1} + 7*16^{0} + 12*16^{-1}$
= 160 + 7 + 0.75
= 167.75

5. **Correct Answer: A**

The contents of the queue are traced as follows.

QUEUE(A);	A
QUEUE(B);	AB
QUEUE(D);	ABD
DEQUEUE();	BD
ENQUEUE(F);	BDF
DEQUEUE();	DF
ENQUEUE(C);	DFC

6. **Correct Answer: D**

a) Represents a post-order traversal.
b) Represents an in-order traversal.
c) Represents a level order traversal.
e) Represents an alphabetic order traversal which would be difficult to accomplish using this type of structure.

7. **Correct Answer: B**

Three of the answers actually provide correct truth values a, b, and d. However, only b is minimal. C'D' covers the column that is bold. The upper left corner that is italicized is A'C'. A'C'D covers two of the elements from A'C' that were not covered by C'D', but in order to obtain the minimal value, the entire block of four must be covered. Lastly, the items with the yellow background finish covering the last two 1's in a minimal manner with BD'. This last block can be tricky to see as it spans the boundaries of the left and right edge.

	C'D' 00	C'D 01	CD 11	CD' 10
A'B' 00	***1***	*1*	0	0
A'B 01	***1***	*1*	0	1
AB 11	**1**	0	0	1
AB' 10	**1**	0	0	0

8. **Correct Answer: E**

The largest 8 bit two's complement numbers is 127. The top bit carries with it a negative value. So the 8 bits would carry the values below.
$(-1)*2^7\ 1*2^6\quad 1*2^5\quad 1*2^4\quad 1*2^3\quad 1*2^2\quad 1*2^1\quad 1*2^0$
This number can be calculated directly as below.

$1*(-1)*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0$
-128 + 32 + 2
-94

Another way to convert two's complement numbers is that if they are positive (the leading digit is 0), just treat them as regular binary numbers. If they are negative, invert all bits and add 1 to get the equivalent positive value for the number. This method is used to obtain the same result.

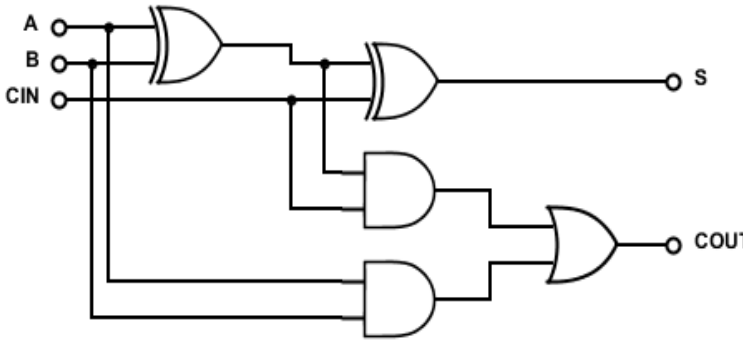
```
 01011101
+00000001
 01011110
```

01011110 $= 0*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0$
$= 0*128 + 1*64 + 0*32 + 1*16 + 1*8 + 1*4 + 1*2 + 0*1$
$= 64 + 16 + 8 + 4 + 2$
$= 94$

Then recall that the number is negative, so -94.

9. **Correct Answer: A**

The circuit provided in the question is a two bit adder with no carry for the input. S provides the output of the first bit and C is the carry output. The first line should have both S and C as F.



The circuit to the right is a two bit adder with carry input. The circuit provided can be used for the first bit if an adder with the C being fed into the two bit adder with carry input. The carry of this second stage can then be fed into a third stage and so on to provide a ripple adder circuit to add larger integers

10. **Correct Answer: E**

The order of the constructors is given below.

```
construct person p
construct person list[0]        Since each student IS-A person
construct student list[0]
construct person list[1]
construct student list[1]
construct person list[2]
construct student list[2]
construct person list[3]
construct student list[3]
```

For a total of 9 constructors.

11. **Correct Answer: C**

Line a is correct, but since it uses an invalid age, the method will set the age to 1. Lines on b and d are correct as is e. The last line, e, uses the overload operator that it inherits from the base person class to increment the age. Line c however may not use the getGpa method as a person does not have access to any of the student methods.

12. **Correct Answer: C**

The student is built upon the base person class. Thus, student IS-A person and inherits all of the public portions of the base person class. When one class just uses an instance of another class, it is said that this class HAS-A instance of the other class.

13. **Correct Answer: D**

The code provided overloads the ++ and > operator. This is an example of polymorphism as the operators will now take on a different behavior for the person class than they would for other data types such as ints, floats or strings.

14. **Correct Answer: A**

The two common protocols in the transport layer of the TCP/IP networking model are User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).
TCP is a connection-oriented protocol and ensures reliable delivery through sequencing and error detection and correction. TCP is used with applications that have an expectation of data being delivered successfully and in the correct order.
On the other hand, UDP is a connection-less protocol and is considerably less sophisticated than TCP. However, this lack of sophistication makes UDP more efficient. Additionally, applications that steam video usually don't want the error detection and correction offered by TCP.
The Internet Protocol (IP), Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP) are network layer protocols. IP is used for addressing and routing on the Internet. ICMP reports the success or failure of data delivery. ARP is used on IPv4 networks to obtain MAC addresses of computers.

15. **Correct Answer: B**

Network Address Translation (NAT) is used in IPv4 networks to allow hosts with private IP addresses to access the Internet.
The Domain Name System (DNS) is used to translate URLs to IP addresses.
A proxy server acts as an intermediary for requests from clients seeking resources from other servers.
Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission.
TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet.

16. **Correct Answer: D**

The core principles of information security are confidentiality, integrity, and availability, often coined as CIA.

17. **Correct Answer: D**

A neural network is modeled after the biological reasoning process that humans possess.
An expert system accumulates knowledge on a particular subject, including conditions and outcomes.
Decision support systems provide assistance to managers in evaluating and choosing among different courses of action.
A transaction processing system (TPS) is the basic business system that assists in making structured decisions.

18. **Correct Answer: B**

This recursive function has a default value of 4 used for the input variable if none is provided for the argument that is passed by value. The base case where it terminates is when the input value is 1 or less.

19. **Correct Answer: C**

The trace follows:

func(8)	8 + func(6)
func(6)	6 + func(4)
func(4)	4 + func(2)
func(2)	2 + func(0)
func(0)	returns 1 to func(2)
func(2)	returns 3 to func(4)
func(4)	returns 7 to func(6)
func(6)	returns 13 to func(8)
func(8)	returns 21 to the cout statement
cout	displays 21

20. **Correct Answer: A**

`pdata` is declared as a pointer to a double and `data[]` is an array of doubles. The assignment statement on line 4 sets `pdata` to the address of the array `data`, which also the address of the first element in the array, `data[0]`. When the `cout` statement is executed, the contents of `data[0]` will be output because the de-referencing operator, *, was used.

21. **Correct Answer: C**

On line 6, the pointer `pdata` is set to the address of `data[2]`. When the `cout` statement is executed, the contents of `data[2]` will be output because the de-referencing operator, *, was used.

22. **Correct Answer: E**

The trace of the code follows:

Line # of code	Code	numbs []	func5 ()	x[]	n	xx	i
8	`int numbs[] = {3,4,5,6};`	{3,4,5,6}	--	--	--	--	--
9	`cout << func5(numbs,4);`	{3,4,5,6}	--	--	--	--	--
1	`int func5(int x[], int n)`	{3,4,5,6}	--	{3,4,5,6}	4	--	--
2	`    int xx = 0;`	{3,4,5,6}	--	{3,4,5,6}	4	0	--
3	`    for(int i = 1; i < n; i++)`	{3,4,5,6}	--	{3,4,5,6}	4	0	1
4	`        xx += x[i];`	{3,4,5,6}	--	{3,4,5,6}	4	4	1
3	`    for(int i = 1; i < n; i++)`	{3,4,5,6}	--	{3,4,5,6}	4	4	2
4	`        xx += x[i];`	{3,4,5,6}	--	{3,4,5,6}	4	9	2
3	`    for(int i = 1; i < n; i++)`	{3,4,5,6}	--	{3,4,5,6}	4	9	3
4	`        xx += x[i];`	{3,4,5,6}	--	{3,4,5,6}	4	15	3
3	`    for(int i = 1; i < n; i++)`	{3,4,5,6}	--	{3,4,5,6}	4	15	4
5	`    return xx;`	{3,4,5,6}	--	{3,4,5,6}	4	15	--
9	`cout << func5(numbs,4);`	{3,4,5,6}	15	--	--	--	--
10	`cout << (sizeof numbs)/(sizeof numbs[0]);`	{3,4,5,6}	--	--	--	--	--

23. **Correct Answer: B**

The sizeof operator returns the size of the operand. In C++, the size of an integer is 4 bytes. `(sizeof numbs)` resolves to 16 and `(sizeof numbs[0])` resolves to 4. And `(sizeof numbs)/(sizeof numbs[0])` resolves to 16/4, or 4.

24. **Correct Answer: C**

The += is a compound assignment operator that modifies the current value of a variable by adding the operand to it.

25. **Correct Answer: E**

The trace of the code follows. Take note: the structure y is passed by reference to both func1 and func2. Therefore, any modifications to the structure a within those functions will actually be modifying y.

Line # of code	Code	y.l	y.t	y.r	y.b	x	y	func 1()	ll	ww
10	`R y;`	--	--	--	--	--	--	--	--	--
11	`y.l = 70;`	70	--	--	--	--	--	--	--	--
12	`y.t = 10;`	70	10	--	--	--	--	--	--	--
13	`y.r = y.l + 25;`	70	10	95	--	--	--	--	--	--
14	`y.b = 30;`	70	10	95	30	--	--	--	--	--
15	`cout << func1(y);`	70	10	95	30	--	--	--	--	--
22	`return (a.r - a.l)*(a.b - a.t);`	70	10	95	30	--	--	--	--	--
15	`cout << func1(y);`	70	10	95	30	--	--	500	--	--
16	`func2(y, 10, 90);`	70	10	95	30	--	--	--	--	--
24	`void func2(R& a, int x, int y) {`	70	10	95	30	10	90	--	--	--
25	`int ll = a.r - a.l;`	70	10	95	30	10	90	--	25	--
26	`int ww = a.b - a.t;`	70	10	95	30	10	90	--	25	20
27	`a.l = x;`	10	10	95	30	10	90	--	25	20
28	`a.t = y;`	10	90	95	30	10	90	--	25	20
29	`a.r = x + ll;`	10	90	35	30	10	90	--	25	20
30	`a.b = y + ww;`	10	90	35	110	10	90	--	25	20
31	`return;`	10	90	35	110	10	90	--	25	20
17	`cout << y.l << "," << y.t << "," << y.r << "," << y.b;`	10	90	35	110	--	--	--	--	--
18	`cout << func1(y);`	10	90	35	110	--	--	--	--	--
22	`return (a.r - a.l)*(a.b - a.t);`	10	90	35	110	--	--	--	--	--
18	`cout << func1(y);`	10	90	35	110	--	--	500	--	--

26. **Correct Answer: A**

See the trace above.

27. **Correct Answer: E**

See the trace above.

28. **Correct Answer: B**

Lines 7 & 8 are function prototypes. A function prototype is used to provide the compiler the information it needs to ensure that the function is used properly. A function prototype must be provided if the function is not defined before it is used.

29. Correct Answer: C

SOLUTION:

This code uses a template function, which alleviates the need to have multiple functions for different types with basically the exact code. The purpose of `func1` in this problem is to find the maximum value in an array. The first call to func1 on line 11 will generate a copy of `func1` for type `int`. The call on line 12 will generate a copy of `func1` for type `char`.

The trace of the code follows:

Line # of code	Code	f[]	g[]	x[]	n	xx	i	x[i]
9	`int f[] = {1,24,34,22};`	{1,24,34,22}	--	--	--	--	--	--
10	`char g[] = {'a','b','c','d'};`	{1,24,34,22}	{'a','b','c','d'}	--	--	--	--	--
11	`cout << func1(f, 4);`	{1,24,34,22}	{'a','b','c','d'}	--	--	--	--	--
1	`template<class T> T func1(T x[], int y){`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	--	--	--
2	`T xx = x[0];`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	1	--	--
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	1	1	24
4	`if(xx<x[i])`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	1	1	24
5	`xx = x[i];`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	24	1	24
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	24	2	34
4	`if(xx<x[i])`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	24	2	34
5	`xx = x[i];`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	34	2	34
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	34	3	22
4	`if(xx<x[i])`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	34	3	22
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	34	5	22
6	`return xx;`	{1,24,34,22}	{'a','b','c','d'}	{1,24,34,22}	4	**34**	--	--
12	`cout << func1(g, 4);`	{1,24,34,22}	{'a','b','c','d'}	--	--	--	--	--
1	`template<class T> T func1(T x[], int y){`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	--	--	--
2	`T xx = x[0];`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	a	--	--
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	a	1	b
4	`if(xx<x[i])`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	a	1	b
5	`xx = x[i];`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	b	1	b
3	`for(int i=1; i<y; i++)`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	b	2	c
4	`if(xx<x[i])`	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	b	2	c

5	xx = x[i];	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	c	2	c
3	for(int i=1; i<y; i++)	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	c	3	d
4	if(xx<x[i])	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	c	3	d
5	xx = x[i];	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	d	3	d
3	for(int i=1; i<y; i++)	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	d	5	d
6	return xx;	{1,24,34,22}	{'a','b','c','d'}	{'a','b','c','d'}	4	**d**	--	--

30. Correct Answer: D
SOLUTION:
See trace above.