2018 Academic Challenge

COMPUTER SCIENCE TEST – STATE

– This Test Consists of 30 Questions –

Computer Science Test Production Team
Kevin Schaefer, Independent Consultant – Author & Reviewer
Sahid L. Rosado Lausell, WYSE – Coordinator of Test Production

GENERAL DIRECTIONS

Please read the following instructions carefully. This is a timed test; any instructions from the test supervisor should be followed promptly.

The test supervisor will give instructions for filling in any necessary information on the answer sheet. Most Academic Challenge sites will ask you to indicate your answer to each question by marking an oval that corresponds to the correct answer for that question. One oval should be marked to answer each question. Multiple ovals will automatically be graded as an incorrect answer.

Be sure ovals are marked as ● , not ⊙ , ⊕ , ◑ , etc.

If you wish to change an answer, erase your first mark completely before marking your new choice.

You are advised to use your time effectively and to work as rapidly as you can without losing accuracy. Do not waste your time on questions that seem too difficult for you. Go on to the other questions, and then come back to the difficult ones later if time remains.

***** Time: 40 Minutes *****

DO NOT OPEN TEST BOOKLET UNTIL YOU ARE TOLD TO DO SO!

©2018 Worldwide Youth in Science and Engineering

"WYSE", "Worldwide Youth in Science and Engineering" and the "WYSE Design" are service marks of and this work is Copyright ©2018 by the Board of Trustees of the University of Illinois at Urbana – Champaign.
All rights reserved

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

1. Which of the following statements is false regarding computer storage?

   A. Hard disk drives and solid-state drives are examples of persistent storage
   B. CPU registers generally hold a small amount of data, but have much faster access times than other types of storage
   C. CPU registers, cache memory, and random-access memory are examples of volatile memory
   D. Random access memory is generally slower than hard disk drives due to the fact that data can only be stored in random locations, and thus requires more time to find data
   E. A computer usually has several layers of cache memory, including a hard drive cache and cache within a CPU

2. Which of the following subnet masks denotes a class A network?

   A. 255.0.0.0
   B. 255.255.0.0
   C. 255.255.255.0
   D. 255.255.255.172
   E. 255.255.255.255

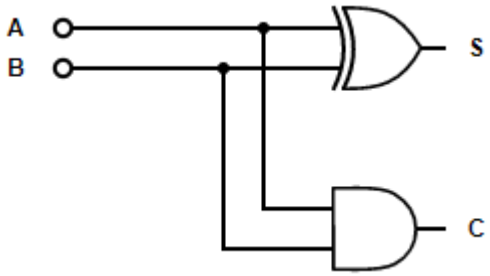
3. Suppose a computer has an IP address of 10.206.0.30, and a subnet mask of 255.255.255.240 and is initiating a TCP connection to a second computer. Which IP address below would the second computer need for that TCP connection to be routed through the first computer's gateway?

   A. 10.206.0.1
   B. 10.206.0.29
   C. 10.206.0.17
   D. 127.0.0.1
   E. All of the above

4. If a computer represents memory addresses with 4 bytes, what is the total number of addressable memory locations?

   A. 4
   B. 2^4
   C. 2^{32}
   D. 10^4
   E. 10^{32}

5. Which of the following truth tables is logically equivalent to the following circuit:



A.

A	B	S	C
F	F	F	F
F	T	T	F
T	F	T	F
T	T	T	T

B.

A	B	S	C
F	F	F	T
F	T	T	T
T	F	T	T
T	T	T	F

C.

A	B	S	C
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T

D.

A	B	S	C
F	F	F	F
F	T	T	T
T	F	T	T
T	T	F	T

E.

A	B	S	C
F	F	T	F
F	T	F	F
T	F	F	F
T	T	F	T

6. Which of the following is logically equivalent to: $\neg\big(x \wedge (x \vee y)\big) \vee x$

A. true
B. false
C. $\neg x \vee \neg(x \vee y) \vee x$
D. $\neg x \wedge \neg(x \vee y) \vee x$
E. Both A and C

7. Which of the following is an equivalent expression if z is false:

$$x \vee (\neg y \wedge z) \vee (\neg x \wedge z)$$

A. x
B. $x \vee \neg(y \wedge z) \wedge \neg(x \wedge z)$
C. true
D. false
E. $\neg x$

8. Which expression represents the minimum sum of products (SOP) of the following Karnaugh Map:

	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

A. A'D' + ABD' + A'B + ABC'D
B. BC' + ABCD' + A'BC + A'B'D'
C. A'B + ABD' + ABC'
D. A' + ABD' + ABC'
E. A'D' + BD' + BC' + A'B

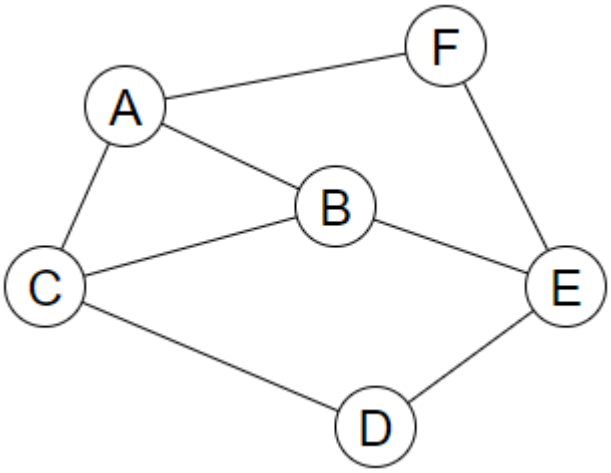
9. Which of the following is true regarding trees and graphs?

A. Trees are a type of graph, but are restricted to having no more than two child nodes
B. A graph may contain cycles, but a tree cannot contain any cycles
C. Both trees and graphs can contain cycles
D. A binary search algorithm is effective at searching a graph
E. None of the above are correct

10. If an algorithm has a time complexity of $2^n + n^2 + \log(n) + n$, what is it's time complexity expressed in Big-Oh notation?

A. $O(n^2)$
B. $O(\log(n))$
C. $O(2^n)$
D. $O(n!)$
E. $O(n)$

11. Which of the following adjacency matrices represents the following graph?



A.

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

B.

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

C.

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

D.

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

E.

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

Use this code for Questions 12, 13, 14, and 15:

```
class A
{
      private:
            int a_private;
      protected:
            int a_protected;
      public:
            int a_public;
};

class B : protected A // Question 14, 15
{
      private:
            int b_private;
      protected:
            int b_protected;
      public:
            int b_public;
};

class C : public B
{
      private:
            int c_private;
      protected:
            int c_protected;
      public:
            int c_public;
};

int main()
{

      C c_inst;

      return 0;
}
```

12. Within the context of `main`, which members of `c_inst` are accessible?

    A. `c_protected`
    B. `b_public`
    C. `a_public`
    D. `a_protected`
    E. Both B and C

13. Within class `C`, which class properties are available?

    A. `a_public`
    B. `a_protected`
    C. `a_private`
    D. `b_private`
    E. Both A and B

14. If the code on the line marked with `// Question 14, 15` were changed to "`class B : private A`", which class properties would be available within class `C`?

    A. `a_public`
    B. `a_protected`
    C. `a_private`
    D. `b_protected`
    E. Both A and B

15. The code on the line marked with `// Question 14, 15` represents which of the following principles of object oriented programming?

    A. Polymorphism
    B. Abstraction
    C. Data-hiding
    D. Inheritance
    E. Principle of least privilege

Use this code for Questions 16 and 17:

```
int size_x = 3, size_y = 5;
int z[size_y];
int c[size_y][size_x] = {
        { 1, 2, 3 },
        { 2, 3, 4 },
        { 3, 4, 5 },
        { 4, 5, 6 },
        { 5, 6, 7 }
};

for(int y = 0; y < size_y; y++)
{
        int a = 0;

        for(int x = 0; x < size_x; x++)
        {
                a += c[y][x];
        }

        z[y] = a / size_x;
}

for(int i = 0; i < size_y; i++)
{
        cout << z[i] << " ";
}
```

16. What will be printed to standard output?

    A. 3 4 5
    B. 6 5 4 3 2
    C. 2 3 4 5 6
    D. 5 4 3
    E. Segmentation Fault

17. What is the time complexity of the code expressed as Big-Oh notation?

    A. $O(3n)$
    B. $O(2^n)$
    C. $O(\log(n))$
    D. $O(3^n)$
    E. $O(n^2)$

18. Which of the following are valid ways to return an `int` array with C++?

    A. `int[] func()`
    B. `int& func()`
    C. `int func()`
    D. `int* func()`
    E. Arrays can only be passed by reference as parameters

Use this code for Questions 19 and 20:

```
int main()
{
        int a = 4;
        int b = 320;
        a = a ^ b;
        b = a ^ b; // Question 20
        a = a ^ b;

        cout << "a: " << a << ", b: " << b << endl;
        return 0;
}
```

19. What will be printed to stdout?

    A. a: 4, b: 320
    B. a: 320, b: 4
    C. Arithmetic overflow
    D. a: 324, b: 324
    E. a: 0, b: 0

20. If the code on the line marked with `// Question 20` were changed to `b = b ^ a;`, what would be printed?

    A. a: 4, b: 320
    B. Arithmetic overflow
    C. a: 324, b: 324
    D. a: 0, b: 0
    E. a: 320, b: 4

Use the following code for Questions 21, 22, 23, 24, and 25:

```
// Calculate the min of three values
int min3(int a, int b, int c)
{
     return min(a, min(b, c));
}

int distance(const char *s, int len_s, const char *t, int len_t)
{
     int cost;

     if(len_s == 0) return len_t;  // Question 24
     if(len_t == 0) return len_s;  // Question 22

     if(s[len_s - 1] == t[len_t - 1])
          cost = 0;
     else
          cost = 1;

     return min3(distance(s, len_s - 1, t, len_t    ) + 1,
                 distance(s, len_s    , t, len_t - 1) + 1,
                 distance(s, len_s - 1, t, len_t - 1) + cost);
}
```

21. What will be printed with this input: `distance("he", 2, "h", 1)`?

    A. 0
    B. 1
    C. 2
    D. 3
    E. None of the above

22. What is the code marked with `// Question 22` called?

    A. Base cases
    B. Pre-test conditionals
    C. Bounds guards
    D. Recursive calls
    E. None of the above

23. What will be printed with this input: `distance("Go!", 3, "", 0)`?

    A. 0
    B. 1
    C. 2
    D. 3
    E. None of the above

24. What would happen if the conditional on the line marked with `// Question 24` were removed?

    A. A compile-time error would be generated
    B. A runtime error would sometimes be generated
    C. A runtime error would always be generated
    D. The program would compile, but would no longer guarantee a correct result
    E. Answers B and D are correct

25. What is the purpose of declaring the `s` and `t` parameters to the distance function as `const`?

    A. `char` pointers are required to be declared as `const` in function parameters
    B. Using `const` prevents the pointers from being modified within the function
    C. The `const` keyword is not needed in this example because the pointers are modified within the function
    D. Both A and B
    E. None of the above

Use the following code for Questions 26, 27, and 28:

```
template <class T>
struct node
{
	node* prev;
	node* next;
	T data;
};

class person
{
	public:
		int Age;
};

int main()
{
	node<person> *head = new node<person>();
	head->data.Age = 40;

	head->prev = NULL;
	head->next = new node<person>();
	head->next->data.Age = 30;
	head->next->prev = head;


	node<person> *tmp = head;

	for(node<person> *tmp = head; tmp != NULL; tmp = tmp->next)
	{
		cout << tmp->data.Age << endl;
	}
	return 0;
}
```

26. What type of the data structure is being generated?

    A. Doubly-linked list
    B. Queue
    C. Stack
    D. Binary Search Tree
    E. Both A and D

27. What code correctly inserts a new node between the two nodes already defined?

A.
```
head->next->prev = new node<person>();
head->next->prev->next = head->next;
head->next->prev->prev = head;
head->next = head->next->prev;
head->next->data.Age = 35;
```

B.
```
head->next = new node<person>();
head->next->prev = head;
head->next->next = head->next;
head->next->data.Age = 35;
```

C.
```
head->next = new node<person>();
head->next->next->prev = head->next;
head->next->prev = head;
head->next->data.Age = 35;
```

D.
```
head->next->prev = new node<person>();
head->next = head->next->prev;
head->next->prev = head;
head->next->data.Age = 35;
```

E. All of the above will correctly insert a new node between the two existing nodes

28. What code will add up the Age property of each item in the data structure?

A.
```
int sum = 0;
node<person> *ptr = head;
while(ptr) {
	sum += ptr->Age;
	ptr = ptr->next;
}
```

B.
```
int sum = 0;
node<person> *ptr = head;
while(ptr->next) {
	sum += ptr->data->Age;
	ptr = ptr->next;
}
```

C.
```
int sum = 0;
node<person> *ptr = head;
while(ptr) {
	sum += ptr->data.Age;
	ptr = ptr->next;
}
```

D.
```
int sum = 0;
node<person> *ptr = head;
do {
	sum += ptr->data.Age;
} while(ptr->next);
```

E.
```
int sum = 0;
node<person> *ptr = head;
for(; ptr->next;
	ptr = ptr->next)
{
	sum += ptr->data.Age;
}
```

Use the following code for Questions 29 and 30:

```
template<class T>
T fun(T x[], int size)
{
	T a = x[0];

	for(int i = 1; i < size; i++)
		if(a > x[i])
			a = x[i];
	return a;
}

int main()
{
	int arr1[] = { 432, 88, 19, 77, 500 };
	int arr2[] = {   0, 30, 22, 17, -2  };

	cout << fun(arr1, 5) << endl;
	cout << fun(arr2, 5) << endl;

	return 0;
}
```

29. What will be printed out with the function call `fun(arr1, 5)`?

    A. 432
    B. 88
    C. 19
    D. 77
    E. 500

30. What will be printed out with the function call `fun(arr2, 5)`?

    A. 0
    B. 30
    C. 22
    D. 17
    E. -2