All multiple choice questions are equally weighted. You can generally assume that code shown in the questions is intended to be syntactically correct, unless something in the question or one of the answers suggests otherwise.

1. **Suppose v is the value of a node N in a binary search tree. Assume that there is a value higher than v in the tree. Among all the values in the tree, where is that next larger value located?**

   A. somewhere in the left subtree of N
   B. in the right child of N (always)
   C. always somewhere in the right subtree of N, athough not necessarily in the right child itself
   D. in the parent node of N (always)
   E. sometimes in the right subtree of N, sometimes in the parent of N

2.
```
class A {
   // ...
};

class B : public A {
   // ...
};

class C : public A {
   // ...
};

A* ap = new A;
B* bp = new B;
C* cp = new C;

int main() {
   // Which of the following assignments raises a COMPILE-time error?

   ap = bp;        // I
   *ap = *cp;      // II
   bp = cp;        // III

   return 0;
}
```

   A. I only
   B. II only
   C. III only
   D. II and III only
   E. none

3.
```
void foo(int N) {
   int temp = 0;
   for (int i = 0; (i < N) && (temp < N); i++)
      for (int j = 0; j < N; j++)
         temp++;
}
```

Chose the correct statement. (hint: First figure out what the function does, then analyze its complexity.)

A. If f(N) represents the running time function of foo, then f(N) = 2 + N(4 + 3N). (The constants 2, 4 and 3 could be off by 1 or 2.)
B. The asymptotic complexity of foo does not depend on N, i.e. it is O(1) (constant).
C. The asymptotic complexity of foo is O(N).
D. The asymptotic complexity of foo is $O(N^2)$.
E. The asymptotic complexity cannot be analyzed without knowing the value of N.

4.
```
#include <iostream.h>

class Base {
public:
  virtual void foo() { bar(); }
protected:
  virtual void bar() { cout << "base "; }
};

class Derived : public Base {
public:
  virtual int foo(int x) { return x; }
protected:
  virtual void bar() { cout << "derived "; }
};

void
main()
{
  Base *basePtr1 = new Base();
  Base *basePtr2 = new Derived();
  basePtr1->foo();
  basePtr2->foo();
}
```

Given the program above, choose the statement which is true:

A. The program outputs "base base ".
B. The program outputs "derived derived ".
C. The program outputs "base derived ".
D. The program won't compile, because Derived does not define void foo().
E. The program won't compile, because method call basePtr2->foo() is illegal.

5. When an array is to be sorted, it may happen that some data values start out being in the same position where they should end up. For example, in the array which is originally

40 -1 33 0

the 33 is right where it will be in the final sorted output:

-1 0 33 40

But as a particular sorting algorithm operates, it might (depending on the algorithm) move such an element out of the position where it belongs (of course, it will eventually get moved back).

Which of the following statements are true?

I. Mergesort never (even temporarily) moves such an element.
II. Quicksort never (even temporarily) moves such an element.
III. Selection sort never (even temporarily) moves such an element.

A. I only
B. II only
C. III only
D. I and III
E. II and III

6. Suppose that the following is from a correct C++ program:

```
bp = new bicycle("built for two", 2, 4);
```

What can be deduced from this?

I. bicycle is the name of a concrete (non-abstract) class.
II. bp was declared as a pointer to bicycle or to some base class of bicycle.
III. bp was declared as a pointer to bicycle or to some class derived from bicycle.

A. I only
B. II only
C. III only
D. I and II only
E. I and III only

7. Two algorithms, A and B, which solve the same problem, are analyzed, and it is found that B's asymptotic running time is greater than A's. A and B are then implemented (carefully and correctly) as programs and tested on various data sets. What is the best prediction?

I. A will always run faster than B.
II. B will always run faster than A.
III. As the size of the test data increases, A is more and more likely to run faster than B.
IV. As the size of the test data increases, B is more and more likely to run faster than A.

A. I only
B. II only
C. III only
D. IV only
E. No predictions are possible without knowing the make and model of the computer used for testing.

8.
```
int findMax(int array[], int arraySize)
{
   int max = array[0];

   for (int i = 1; i < arraySize; i++) {
      if (max < array[i])
         max = array[i];
   }

   return max;
}
```

This function is intended to find the largest value in an array of integers, under the precondition arraySize >=1. Suppose an additional precondition for this function is given as array[i] <= array[i+1] for all i from 0 to arraySize-2.

What is true about the function, as coded?

I. The function returns the correct answer only for those arrays which meet the new precondition.
II. The function returns the correct answer faster for arrays which meet the new precondition.
III. There is a much faster algorithm which returns the correct answer for all arrays meeting both preconditions.

A. I only
B. II only
C. III only
D. I and II only
E. II and III only

9. In the following code fragment, Q is a queue of integers:

```
Q.insert(1);

while ( !Q.isEmpty() ) {
   int f = Q.getFront();
   if (f > 10)
      f = Q.dequeue();
   else
      Q.enqueue(f+1);
}
```

The last time through this loop, what value is removed from the queue?

A. 1
B. 9
C. 10
D. 11
E. can't tell---infinite loop

10. Suppose a tree has 7 nodes and a height of 3. (Our definition has the root at level 1.) What is true?

I. The tree must be a binary tree.
II. The tree cannot be a binary tree.
III. The tree has at least one node at level 3.
IV. At least one node at level 3 is not a leaf.

A. I only
B. II only
C. III only
D. I and IV only
E. II and IV only

11. A function which maps a key to an index or bucket number is called a…

A. accessor method
B. overloaded index function
C. hash function
D. leaky bucket function
E. retrieve function

12. The preorder traversal of a certain binary search tree is

10 5 3 2 15 12 20

If the value 11 is then added to this tree, what is the preorder traversal of the resulting tree?

(hint: First, draw the tree, using the fact that it is a search tree.)

A. 11 10 5 3 2 15 12 20
B. 10 11 5 3 2 15 12 20
C. 10 5 3 2 11 15 12 20
D. 10 5 3 2 15 11 12 20
E. 10 5 3 2 15 12 11 20

13. Rank the following (1-7) into increasing order of growth, with 1 as smallest (slowest growing) and 7 as largest (fastest growing):

constant
cubic
exponential
linear
logarithmic
n log n
quadratic

Having done this, which are the entries with ranks 5 and 6?

A. 5. exponential
6. quadratic
B. 5. quadratic
6. cubic
C. 5. constant
6. quadratic
D. 5. n log n
6. exponential
E. 5. logarithmic
6. cubic

14. In the mergesort algorithm, what is the asymptotic running time of the step of merging sorted subarrays?

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

15. Suppose the partition function (of quicksort) is called on this array:

4 7 6 5 1 8 0

Which of the arrays below could result from the call, i.e. which of them satisfies the postconditions of partition? (Assume partition chooses the first element as the pivot.)

A. 1 0 4 5 6 8 7
B. 4 0 1 5 6 8 7
C. 4 0 1 5 6 7 8
D. 1 0 5 4 6 8 7
E. 4 8 7 6 5 1 0

16. Which of these sorting algorithms has the best (lowest) asymptotic running time when the input list is already sorted?

I. insertion sort
II. selection sort
III. quicksort (choosing the first element of the array as the pivot)
IV. mergesort

A. I
B. II
C. III
D. IV
E. III and IV, with the same asymptotic running time

17. Consider the following definition of class A:

```
class A {
private:
    void foo(int, int) = 0;
public:
   void bar(double x);
   void baz(double x, double y);
   void quax(double xyz[]);
};
```

Which of the following is true about the declaration of foo within A?

A. The declaration is illegal because it does not specify argument names.
B. The declaration is not logical because the method cannot be called by client code.
C. The declaration is illegal because the method is not virtual.
D. The declaration is not logical because the other methods cannot call foo.
E. The declaration may be both legal and logical, depending on the other code in the program.

18. ***Programming Problem*** **(worth 4 multiple choice questions)**

There are many different ways a set of values could be arranged in a binary search tree. Two trees are called "content-equal" if they contain exactly the same node values, regardless of how the nodes are arranged in the tree.

Design and implement the function isContentEq, which should return true if and only if the two given binary search trees are "content-equal." You may assume that neither tree contains duplicate values. Make no assumptions about how large or deep the trees are. You must use the function prototype and tree node definition shown below. You may not modify the trees. If you call any functions, you must provide full code for them.

A. Give an overview (in English) of how your algorithm will operate.

(PROBLEM CONTINUES ON NEXT PAGE)

```
// Use this definition of a tree node (without modification):
struct BinTreeNode {
  double value;
  BinTreeNode *left, *right;
};

// Use this prototype:
bool isContentEq(BinTreeNode *T1, BinTreeNode *T2);
```

B. Implement the function isContentEq. Remember, if you call any helper functions, you must fully declare and implement them.

```
// Use this definition of a tree node (without modification):
struct BinTreeNode {
  double value;
  BinTreeNode *left, *right;
};

// Use this prototype:
bool isContentEq(BinTreeNode *T1, BinTreeNode *T2);
```