You may take this test with you after the test, but you must turn in your answer sheet. This test has 33 multiple-choice questions, each worth 3 points, for a total of 99 points, plus one for free.

This test is worth 20% of your final grade. You must put your answers on the answer form using a #2 pencil. This test is open book and open notes. For the multiple choice problems, select the best answer for each one and select the appropriate letter on your answer sheet. When a problem describes a segment or fragment of code you may assume the rest of the program is correct and would be supplied to make it work. You have two hours.

1. What numbers are in the values array after running the code segment shown below at right?

A) 1 3 4 7 9
B) 9 7 4 3 1
C) 1 3 4 3 1
D) 9 7 4 7 9

const int Max = 5;
int values[Max] = {1,3,4,7,9};
for (int i=0; i<Max; ++i) {
    values[i] = values[Max-i-1];
}

2. What is the output of the following C++ program segment, called with confuseDriver()?

A) s + y = 3
B) s + y = 6
C) s + y = 8
D) s + y = 10

int s=1, y=3;

void confuse1(int y, int s)
{
    s++;
y++;
}

void confuse2(int b, int &s)
{
y = ++(s);
s = b;
}

void confuse3(int &a, int &s)
{
a = s + 1;
s++;
}

void confuseDriver()
{
    int s=2;
    confuse1(s, y);
    confuse2(s, y);
    confuse3(s, y);
    cout << "s + y = " << s+y << endl;
}
3. Consider the following function declaration:
   
   ```cpp
   void setValues( int values[ ]);
   ```
   Note that there is no value inside the square brackets to specify the array size. Which of the following statements is the best description of this situation?

   - A) This will cause a compiler error and will not run
   - B) This will compile, but will cause a run-time error
   - C) Having no number will allow writing past the end of the array, which would be disallowed if a number value were supplied.
   - D) Having no number is not a problem.

4. Consider the following function declaration:
   
   ```cpp
   void displayTable( int values[ ][ ]);
   ```
   Note that there is no value inside either the first or second set of square brackets to specify the array size. Which of the following statements is the best description of this situation?

   - A) Values are not required in either set of brackets
   - B) You must have a value in the first set of brackets, but not necessarily the second
   - C) You don't need a value in the first set of brackets, but do in the second
   - D) You must supply values for both sets of brackets

5. Imagine we create a Date class with the following single constructor:
   
   ```cpp
   Date( int theMonth, int theDay, int theYear) {
       month = theMonth; day = theDay; year = theYear;
   }
   ```
   The driver code then has the following, which generates an error:
   
   ```cpp
   Date d1;
   ```
   What is the likely reason for this error?

   - A) There is no default constructor for the Date class
   - B) The Date class is declared after the declaration of d1
   - C) There is already a declared Date object with the name d1
   - D) This is a declaration and is not calling the Date class constructor

6. Consider using Bubble Sort to put the following elements into ascending order:
   
   7 6 12 2 9 4 3
   
   What of the following best describes the relationship between the data and the passes?

   - A) The number of passes required before the elements are in ascending order is smaller when doing our passes starting from the left.
   - B) The number of passes required before the elements are in ascending order is smaller when doing our passes starting from the right.
   - C) The number of passes for this data in particular is the same regardless of whether we start our passes through the data starting from the left or from the right.
   - D) The number of passes is always the same for all data regardless of starting our passes from the left or from the right.
7. Consider using binary search to find a particular number within a sorted array of 100 unique random numbers. On average how many numbers will have to be examined before the number you are searching for is found? Include the number found in your count.

A) 6  
B) 7  
C) 10  
D) 50  
E) 100

8. What is the output of the code segment shown at right when function testf1() is called?

```cpp
void f1( int y)
{
    static int x=1;
    cout << x++ << " ";
}
void testf1()
{
    for( int x=0; x<3; x++)
    {
        f1( x);
    }
}
```

A) 0 0 0  
B) 1 1 1  
C) 0 1 2  
D) 1 2 3

9. Consider the declaration shown at right. Which of the following would compile and run, allowing storing first name and age properly?

```cpp
class Person {
public:
    Person( string theName, int theAge)
    {
        name = theName;
        age = theAge;
    }
    string name;
    int age;
};
Person p1("Heron", 23);
Person p2("Darin", 37);
Person *pPerson = &p1;
```

```cpp
A) cout << "Enter first name and age: ";
cin >> p1.name >> (*pPerson).age;
B) cout << "Enter first name and age: ";
cin >> name.p1 >> age.(*pPerson);
C) cout << "Enter first name and age: ";
cin >> p1.name >> *(`pPerson.age);
D) cout << "Enter first name and age: ";
cin >> p1.name >> &(pPerson->age);
```
10. What are the results of calling functions $fa$ and $fb$ in the code shown below:

Option A:

```c++
void fa( int x)
{
    for( int i=0; i<x; i++)
    {
        for( int j=0; j<x; j++)
        {
            cout << "*";
        }
    }
    cout << endl;
}
```

Option B:

```c++
void g1( int x) {
    if(x==0) return;
    cout << "*";
    g1(x-1);
}
void g2( int x, int y) {
    if(y==0) return;
    g1( x);
    cout << endl;
    g2(x,y-1);
}
void fb( int x) {
    g2( x,x);
}
```

A) For the same parameter value $fa$ will give more output than $fb$
B) For the same parameter value $fb$ will give more output than $fa$
C) $fa$ will give exactly the same output as $fb$
D) One version will give output, but the other results in an infinite loop

11. Consider the code segment shown at right. After the function call to changeLetters(...) the value of number has changed. What is the most likely cause of this change?

A) number is a global variable instead of a local variable as it should be
B) Although number is not passed to function changeLetters(), function changeLetters() itself calls a second function which changes number
C) There is some ASCII control character that is present in the code even though they are not visible
D) Function changeLetters() overwrites the end of array letters

12. Consider the class shown at right used to implement a list which has a sentinel node, so there is always at least that one node on the list.
Consider the following function declarations used to prepend nodes at the beginning of the list:

I. ```c++
   Node *prepend(int value, Node *pHead)
```
II. ```c++
     void prepend(int value, Node *&pHead)
```
III. ```c++
     void prepend(int value, Node **pHead)
```

How many of the above functions could be used to prepend nodes at the beginning of the list?

A) None
B) 1
C) 2
D) 3
13. Assume you have a Node class and pHead is a pointer to the head of the following list of Nodes:

```
6 5 4 3 2 1
```

What is the output from the statement: `cout << pHead->pNext->pNext->data;`

A) 5  
B) 4  
C) 3  
D) 2

14. What is the best choice to declare the type of a dictionary where you don't know ahead of time the size of the words?

A) char dictionary;  
B) char dictionary[DictionarySize];  
C) char *dictionary;  
D) char *dictionary[DictionarySize];

15. If we knew the number of dictionary words ahead of time and the words were all the same size, what would be the effect of declaring the dictionary as:

```
char dictionary[ numberOfWords * (wordSize+1)];
```

A) It would not be possible to pull out individual words  
B) It would be possible to store words this way, however C string functions could not be used  
C) It would be possible to store words this way, however the whole dictionary would basically be one extremely large C string  
D) It would be possible to store words this way and still use C string functions, doing arithmetic to find the address of each word starting position.

16. What advantage does a linked list have over an array?

A) It uses less memory  
B) Accessing a linked list node is typically faster than accessing an array element  
C) An element can be inserted in the middle without having to shift other elements  
D) All of the above

17. Given the code shown below, what is stored in array letters after calling function Problem17()

```c

int eval( char *pC)
{
    if( *pC == '\0' ) return 0;
    return eval( ++pC) + 1;
}

void modify(char letters[], int index1)
{
    char temp = letters[ index1];
    int index2 = eval( letters) - 1;
    letters[ index1] = letters[ index2];
    letters[ index2] = temp;
}

void f17( char letters[], int index)
{
    int x = eval(letters);
    if( index >= x) return;
    modify( letters, index);
    f17( letters, ++index);
}

void Problem17()
{
    char letters[]="abcd";
    f17( letters, 0);
    cout << letters;
}
```

A) abcd  
B) dcba  
C) cdab  
D) dabc
18. Consider the two approaches shown below to implement a linked list, where we need to both prepend at the beginning of the list, as well as append at the end of the list.

<table>
<thead>
<tr>
<th>Approach A:</th>
<th>Approach B:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram A" /></td>
<td><img src="image2.png" alt="Diagram B" /></td>
</tr>
</tbody>
</table>

Which of the descriptions below is the most accurate?

A) Neither A nor B can be used  
B) Approach A can be used, but B cannot  
C) Approach B can be used, but A cannot  
D) Either A or B can be used, though one approach may be more efficient than the other

19. Consider a version of a linked list where we always add new nodes to the tail of the list and delete existing nodes from the head of the list, like in a line to order food at a fast-food restaurant. Consider the two head-and-tail pointer implementations shown below:

<table>
<thead>
<tr>
<th>Approach I:</th>
<th>Approach II:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Diagram I" /></td>
<td><img src="image4.png" alt="Diagram II" /></td>
</tr>
</tbody>
</table>

Which of these two approaches is preferable?

A) Neither one would work correctly  
B) Approach I is preferable  
C) Approach II is preferable  
D) Neither one is preferable, they are both equally well suited

20. What is the output of the function shown at right, when the input is: cat$

   A) cat  
   B) cat$  
   C) tac  
   D) tac$

21. What is the output of the function shown at right, when the input is: cat$

   A) cat  
   B) cat$  
   C) tac  
   D) tac$
For the next four problems consider the code shown at right, similar to the maze program discussed in class. The code is called using: `makeMove( start);`

22. What is the result of running this program relative to the values in the `moves` array?

   A) No matter the `moves` order, it would always still find the solution
   B) It would find the solution when using some `moves` orders, but not others.
   C) It can only find the solution using one `moves` order
   D) No `moves` order will allow it to find the solution using the code as shown.

23. For this maze what is the relationship between the ordering of the values in the `moves` array and the `start` and `goal` values?

   A) If an ordering of the `moves` array works, then it will still work if the `start` and `goal` values are reversed.
   B) If an ordering of the `moves` array works, then it will **not** work if the `start` and `goal` values are reversed.
   C) An ordering of the `moves` array will work for both the original and reversed `start` and `goal` values only if we always do the *left* and *right* moves before doing the *down* and *up* moves.
   D) An ordering of the `moves` array will work for both the original and reversed `start` and `goal` values only if we always attempt to do the *down* and *up* moves before doing the *left* and *right* moves.

24. What would be the result if we set `movesSize = 8;` and we set the moves array to be:
   `{11,1,-9,-10,-11,-1,9,10}`

   A) The results of running the program would be unchanged
   B) A solution would still be found, but it would take more steps
   C) A solution would still be found, but it would take fewer steps
   D) A solution would no longer be found.

25. What would be the result in the original code above if we uncomment the line of code at // Point A

   A) No matter the `moves` order, it would always still find the solution
   B) It would find the solution when using some `moves` orders, but not others.
   C) It can only find the solution using one `moves` order
   D) No `moves` order will allow it to find the solution using the code as shown.
For the following three linked list questions, assume we have already implemented the following functions, using the same Node structure we discussed in class. Assume list values are randomized to start.

I. Node *findLastNode( Node *pHead)  // find last node on list
II. void prepend(Node *&pHead, Node *pNew) // put new node at list front
III. void insertInOrder(Node *&pHead, Node *pNew) // insert in ascending order
IV. bool isInList(int data, Node *pHead) // return true if data is in list
V. bool sameAsNext(Node *pTemp)         // true if node data is same as next's
VI. Node *reverseList( Node *pHead)      // reverse the list

26. Which of the above functions would be best to use with minimum additional code to implement:
   void concatenate(Node *&pListA, Node *pListB);
   which should result in all the nodes of list B being added to the end of list A:
   
   A) This can be done easily without using any of the above functions
   B) prepend
   C) insertInOrder
   D) findLastNode

27. Which of the above functions would be best to use to implement:
   void sortDescending(Node *&pHead);
   where the resulting code run time is in proportion to the list length?
   
   A) findLastNode, prepend
   B) isInList, prepend
   C) insertInOrder, findLastNode
   D) insertInOrder, prepend

28. Which set of the above functions would be best to use with minimum additional code to implement:
   void removeDuplicates(Node *&pHead);
   
   A) findLastNode, prepend
   B) findLastNode, insertInOrder
   C) sameAsNext, prepend
   D) sameAsNext, insertInOrder, prepend

29. What is the best description of the function shown at right?
   
   A) It does not compile
   B) It compiles but gives a run-time error
   C) It deletes all the nodes on the list
   D) It traverses the list and deletes the last node

30. What is the best description of the function shown at right?
   
   A) It does not compile
   B) It compiles but gives a run-time error
   C) It deletes all the nodes on the list
   D) It traverses the list and deletes the last node
For the next three problems carefully consider the C/C++ program segment given below, called with:
function doit(). Note the lines Position A and Position B which are referred to in the questions below. Assume input to the program is: 1 3 5 7 -1

```c
struct Node {
    int data;
    Node *pNext;
};

void display( Node *pHead)
{
    while( pHead != NULL) {
        cout << pHead->data << " ";
        pHead = pHead->pNext;
    }
}

Node * modify( Node *pHead)
{
    Node *pTemp;
    if (pHead->pNext == NULL)  {
        return pHead;
    } else {
        pTemp = modify(pHead->pNext);
        pHead->pNext->pNext = pHead;
        // Position B left
        return pTemp;
    }
}

void doit()
{
    int number = 0;
    Node *pHead = NULL;
    Node *pTemp;
    cout <<"Enter numbers, then -1: ";
    // prepend nodes
    while ( number != -1) {
        cin >> number;
        if (number != -1) {
            pTemp = new Node;
            pTemp->data = number;
            pTemp->pNext = pHead;
            pHead = pTemp;
        }
    }
    pTemp = pHead;
    pHead = modify( pHead);
    // Position A right
    display( pHead);
}
```

31. What is the output of the code as shown?

A) 7 5 3 1  
B) 1 3 5 7  
C) Only part of the input  
D) It compiles and runs, but goes into an infinite loop

32. What is the output of the original code if we replace only the line of code at Position A right above with:  
pTemp->pNext = NULL;

A) 7 5 3 1  
B) 1 3 5 7  
C) Only part of the input  
D) It compiles and runs, but goes into an infinite loop

33. Starting again with the unchanged original code, what is the output if we replace the line of code at Position B left above with:

```c
pHead->pNext = NULL;
```

A) 7 5 3 1  
B) 1 3 5 7  
C) Only part of the input  
D) It compiles and runs, but goes into an infinite loop