True (A)/False(B) (2 pts each):
1. In a function with call-by-reference parameters, the values of the actual arguments are passed to the function.

2. The locations of the various indexed variables in a statically declared array can be spread out all over the memory.

3. The following array declaration is legal in C++
   double scores[]={0.1,0.2,0.3};

4. If you use the const modifier in a function declaration, you do not include it in the function definition.

5. Using the == operator on a string object results in the same value as using strcmp on c-style strings.

6. Every recursive definition may be rewritten iteratively.

7. Recursive functions always execute faster than an iterative function.

8. The following declares a c-string and initializes it to "speaker"
   char str[]="speaker";

9. The following declares a c-string variable that has a total of 12 elements
   char str[]="hello world";

10. If p1 is an integer pointer variable, with the value of 1000, p1++ changes P1 to point to the memory location 1001.

11. When you return a dynamic array to the heap/freestore, you must include the number of elements in the array.

12. You can assign an array to a pointer variable.

13. The size of dynamic arrays must be declared at compile time.

14. If p1 and p2 are both pointers that point to integers in memory, the condition p1==p2 will be true if the values that are in those memory locations are the same.
Multiple Choice (3 pts each):

15. Which of the following will print out the string value pointed to by str?
   char str[30];
   cin >> str;
   a. cout << str;
   b. for(int i=0;i<30;i++)
      cout << str[i];
   c. int i=0;
      while(i<30 && str[i] != '\0')
      cout << str[i];
   d. All of the above
   e. A and B

16. Given an array named scores with 25 elements, what is the correct way to access the 25th element?
   a. scores+25
   b. scores[24]
   c. scores[25]
   d. scores[last]

17. Give the following declarations, which of the following is a legal call to this function?
   int myFunction(int myValue);
   int myArray[1000];
   a. cout << myFunction(myArray);
   b. cout << myFunction(myArray[0]);
   c. myArray = myFunction(myArray);
   d. myArray[1] = myFunction(myArray[0]);
   e. B and D

18. The precondition(s) for a function describe:
   a. What is true after the function is executed
   b. What the function does
   c. How to call the function
   d. What must be true before the function executes

19. What is wrong with the following code?
   float scores[10], total;
   a. Cannot declare regular and array variables together.
   b. Arrays must be integers
   c. The 10 should be replaced with a variable name, whose value is input from the user
   d. Nothing.
20. Which of the following function prototypes are not valid?
   a. `void doSomething(int& x, int y);`
   b. `void doSomething(int& x, int& y);`
   c. `void doSomething(int x, int y);`
   d. all are not valid
   e. all are valid

21. You should make a parameter a reference parameter if:
   a. You need the function to change the value of the argument passed to the function.
   b. You need to be able to change the value of the parameter in the function, but not the value of the argument.
   c. Always.
   d. If any of the other parameters are reference parameters.

22. If you write a function that should use call-by-reference, but forget to include the ampersand,
   a. The program will not compile
   b. The program will not link
   c. The program will not run without a run-time error
   d. The program will run with incorrect results
   e. It doesn't matter

23. The following function definition has an error in it. What line is this error on?
   0. void f1(const double array[], int size)
   1. {
   2.     int i=0;
   3.     while(i< size)
   4.     {
   5.         array[i] += 2;
   6.         cout <<array[i];
   7.         i++;
   8.     }
   9. }
   a. 0
   b. 2
   c. 5
   d. 6

24. Which of the following function declarations will accept the following 2-D array?
   int pages[10][30];
   a. `void f1(int pages[][], int size);`
   b. `void f1(int pages[][30], int size);`
   c. `void f1(int pages[10][], int size);`
   d. `void f1(int& pages, int size);`
25. What is the output of the following function and function call?

```c
void calculateCost(int count, float& subTotal, float& taxCost);

int main() {
    float tax = 0.0, subTotal = 0.0;

    calculateCost(15, subTotal, tax);
    cout << "The cost for 15 items is " << subTotal << ", and the tax for " << subTotal << " is " << tax << endl;
    return 0;
}
```

a. The cost for 15 items is 3.00, and the tax for 3.00 is 0.30;
b. The cost for 15 items is 0.00, and the tax for 3.00 is 0.00;
c. The cost for 15 items is 0.00, and the tax for 3.00 is 0.30;
d. The cost for 15 items is 3.00, and the tax for 3.00 is 0.00;

26. If you want to read into a c-string, you must ensure that the user does not enter more characters than

a. The size of the c-string
b. The size of the c-string + 1
c. The size of the c-string -1
d. It doesn't matter.

27. What is wrong with the following attempted c-string declaration and initialization?

```c
char str1[5]={"a", "b", "c"};
```

a. There are only 3 values in the braces
b. The single quotes should be double quotes
c. The values do not constitute a c-string
d. Nothing

28. How can you assign the value "toaster" to a c-string name str of size 10?

a. str="toaster";
b. str=toaster;
c. strcpy(str,"toaster");
d. str.strcpy("toaster");
29. Which assignment statements will copy the value "toaster" into a C++ string object, str1?
   a. strcpy(str1,"toaster");
   b. str1 = "toaster";
   c. str1 = toaster;
   d. str1 += toaster;

30. Which is the proper way to determine how many characters are in a C++ string object named str, i.e. string str;?
    a. str.getLength()
    b. str.length()
    c. strlen(str)
    d. length(str)

31. Given the following declarations, which of the following is legal syntax?
    string str="Your name";
    char c_string[20]="My name";
    a. str = c_string;
    b. c_string = str;
    c. strcpy(c_string, str.c_str());
    d. strcpy(c_string, str);
    e. A and C

32. Given that p1 is a pointer to a C++ string object, string *p1, which of the following are legal statements?
    a. p1 = new int;
    b. cout << *p1;
    c. p1 = new char[10];
    d. *p1 = new string;
    e. B and D

33. In which case would you consider using a dynamic array?
    a. If the array is small, and the size is known before the program runs.
    b. If the program needs to get the size of the array from the user
    c. If the array size is big, but known at compile time
    d. You should always use a dynamic array

34. Which of the following correctly declares a dynamic array of C++ string objects?
    a. p1 = new string(13);
    b. p1 = new string[];
    c. p1 = new string[13];
    d. p1 = new stringArray(13);
35. What is the output of the following code fragment?
   float *p1;
   p1 = new float;
   *p1 = 3.0;
   cout << *p1;
   a. 3.0
   b. unknown, the address p1 points to is not initialized
   c. unknown, the code is illegal, p1 points to a dynamic array
   d. 0.0

36. Given that p1 is an integer pointer variable, and a1 is a static integer array, which of the
    following statements is not a legal statement?
   a. p1 = a1;
   b. cout << p1[0];
   c. cin >> p1[0];
   d. a1 = p1;

37. Which of the following statements correctly returns the memory to the heap/freestore from
    the dynamic array pointer, p1?
   a. delete [] p1;
   b. delete p1[];
   c. delete *p1;
   d. delete p1;

38. If a program requires a dynamically allocate two-dimensional array, you would allocate the
    memory by using
   a. p1 = new int*[numRows];
      for(int i=0; i < numRows; i++)
      p1[i] = new int[numColumns];
   b. p1 = new int*[numRows][numColumns];
   c. p1 = new[numRows][numColumns]int;
   d. none of the above

Extra Credit (2 pts each):

39. If you try to solve a problem recursively, you should
   a. find all the stopping cases for the recursive definition
   b. find a recursive call that will lead towards one of the stopping cases
   c. all of the above
   d. none of the above
40. What is wrong with the following recursive function? It should print out the array backwards.

```c
void print(int array[], int start, int size)
{
    if(start == size)
        return;
    else
    {
        print(array, start+1, size);
        cout << array[start] << endl;
    }
}
```

a. the cout statement should be before the recursive call  
b. the stopping condition is wrong  
c. the recursive call is wrong  
d. nothing

41. What is the output of the following code fragment?

```c
int f1(int n, int m)
{
    if(n < m)
        return 0;
    else if(n==m)
        return m+ f1(n-1,m);
    else
        return n+ f1(n-2,m-1);
}

int main()
{
    cout << f1(5,4);
    return 0;
}
```

a. 0  
b. 2  
c. 4  
d. 8  
e. infinite recursion
42. What is the output of the following code fragment?
```cpp
int f1(int base, int limit)
{
    if(base > limit)
        return -1;
    else {
        if(base == limit)
            return 1;
        else
            return base * f1(base+1, limit);
    }
}
```
```cpp
int main()
{
    cout << f1(2,4) << endl;
    return 0;
}
```

a. 2
b. 3
c. -1
d. 6

43. What is the output of the following code fragment?
```cpp
int f1(int x, int y)
{
    if(x<0 || y<0)
        return x-y;
    else
        return f1(x-1,y) + f1(x,y-1);
}
```
```cpp
int main()
{
    cout << f1(2,1) << endl;
    return 0;
}
```

a. 0
b. -1
c. 5
d. -5