What is your name?: ___________________________(2 points. Don’t miss this one!)

There are two sections:

I. True / False 20 points; (1 point each, 20 questions)
II. Multiple Choice 78 points; (13 questions, 6 pts. each)

---------------
98 + 2 points for name = 100 points total

This test is worth 15% of your final grade. For all problems you must put your answers on the bubble form using a #2 pencil. This test is open book and open notes, but no calculators. You have 50 minutes.

I. True / False Section: (1 pt. each) Select the best answer to each problem below. Be careful - some are tricky.

- If the answer is True, fill in a on your answer form. If the answer is False, fill in b on your answer form.
- Several problems ask you to determine whether something is valid. Something is valid if it would not generate a compiler error and would execute without the program crashing.

1. T    F       If a loop will always be executed at least once, a while loop is just as good as a do-while loop.
2. T    F       A for loop can be used interchangeably with a while loop. It really comes down to user preference.
3. T    F       Nested if-else statements are just as good as a switch statement when it comes to handling menu options.
4. T    F       The flushout statement is useful for debugging, sending the contents of the output buffer to the screen.
5. T    F       To ensure a character stored in variable c is upper case, we could use the following:
                 if( (c>='a') && (c<='z')) c -= 'a' - 'A';
6. T    F       Decimal value -31 written as an 8-bit twos-complement negative number is 11100001.
7. T    F       The statement
                 scanf("%d %d", intValue1, intValue2);
                 has the same effect as the statement
                 cin >> intValue1 >> intValue2;
8. T    F       To check if character c is a numeric character, we could use the if statement:
                 if( '0' <= c <= '9')
9. T    F       The sscanf statement is just like scanf, except it is used to read in only strings from the keyboard.
10. T    F      Given the declarations
                 int i;
                 char aWord[ 15];
                 int j;
                 If some section of code that writes to array aWord goes past the end of the array, it could attempt to modify either variables i or j, which is not allowed and so would automatically stop the program’s execution.
11. T    F      The strtok function is used to verify that the string total is ok.
For the next 8 problems, consider the declarations shown below. Note that the decimal ASCII value for 'A' is 65.

```
char fullText[] = {"Madam I'm Adam");
char word2[] = {"Madam\0");
char word4[] = {"Adam\0");
char *pWord = word4;
```

12. The statement
   `printf( "%c", fullText);`
   has the same effect as the statement
   `printf( "%c", fullText[0]);`

13. The statement
   `printf( "%d ", fullText[0]);`
   has the same effect as the statement
   `printf( "%d ", fullText);`

14. The statement
   `strcmp( word2, "Madam")`
evaluates to the numerical value of 0.

15. The statement
   `printf( "%s", word4+2);`
   will display the text
   `am`

16. The statement
   `printf( "%c", pWord);`
   will display the text
   `A`

17. The statement
   `printf( "%c", *pWord);`
   will display the text
   `A`

18. The statement
   `printf( "%c", &pWord);`
   will display the text
   `A`

19. The statement
   `printf( "%c", &*pWord);`
   will display the text
   `A`

20. The statements
    *(pWord + 2) = 'd';
    `printf( "%s", pWord);`
    will display the text
    `Addm`
II. Multiple Choice: (6 pts. each) For the following problems, select the best answer for each one and select the appropriate letter on your answer sheet. Be careful - more than one answer may seem to be correct. All of the following programs do compile and run.

21. Consider the program given below.

```c
#include <iostream.h>
int main()
{
    int x;
    cout<"Please enter a multi-digit integer: ";
    cin>> x;
    while (x > 0) {
        cout< x % 10;
        x /= 10;
    }
    cout<"\n";
}
```

Its output can best be described as:

a) The input digits in reverse order  
b) The remainder of the digits after dividing by 10  
c) A tenth of the original input  
d) The input divided by 10 as many times as there are input digits  
e) None of the above

22. Consider the program given below.

```c
#include <iostream.h>
int main()
{
    int i,j;
    for (i=0; i<100; i++)  {
        for (j = 2; j < i; j++)  {
            if ( (i%j) == 0)
                break;
        }
        if (j==i)
            cout< i <" ";
    }
    cout << endl;
}
```

Its output is:

a) Prime numbers less than 100  
b) Odd numbers less than 100  
c) Even numbers less than 100  
d) All numbers less than 100  
e) None of the above
23. Assume you are using a binary search to find a particular number in a sorted list of 2050 numbers stored in an array. What is the maximum number of numbers you would need to look at in order to find that particular name?
   a) 12  
b) 13  
c) 14  
d) depends on the average size of the numbers  
e) None of the above  

24. Consider using a bubble sort (as discussed in class) to sort 21 numbers. What is the maximum number of passes we need to make through the data before the numbers are sorted?
   a) 11  
b) 20  
c) 21  
d) 42  
e) None of the above  

25. The binary number 1010110 when interpreted as a signed binary number is equal in decimal to:
   a) 86  
b) 22  
c) -22  
d) 53  
e) None of the above  

Consider the function given below for the following two problems:

```
int F(int Num1, int Num2)
{
    if (Num1 < Num2)
        return 0;
    else if (Num1 == Num2 - 1)
        return 1;
    else
        return F(Num1 - 1, Num2 + 1) + 2;
}
```

26. The function call F(4, 1) gives as output:
   a) 4  
b) 5  
c) 6  
d) Nothing, the function doesn’t terminate  
e) None of the above  

27. The function call F(10, 4) gives as output:
   a) 0  
b) 3  
c) 8  
d) Nothing, the function doesn’t terminate  
e) None of the above
For the next 2 questions, consider the maze program shown below similar to the one we discussed in class. You may assume that the `cameFrom` array’s values are initially all 0 by default. Note that for this maze the correct solution path in order is: 11 12 22 32 42 43 44 54 64 65 66 76 86 87 88

28. The output of the program as shown above is:

   a) The solution path in order
   b) A path that is not the solution path, since it includes positions not in the path
   c) The solution path, except in reverse order
   d) A bunch of numbers, since it gets stuck in an infinite loop
   e) None of the above

29. Assume the same code as above, only this time `main()` looks like this:

    ```c
    main()
    {
        makeMove( start);
        displaySolution( finish); /* This line changed */
    }
    ```

    The output to this program is:

   a) The solution path in order
   b) A path that is not the solution path, since it includes positions not in the path
   c) The solution path, except in reverse order
   d) A bunch of numbers, since it gets stuck in an infinite loop
   e) None of the above
30. Consider the following function:

```cpp
int F2( int x)
{
    if ( x < 10) {
        return x;
    }
    else
        return F2( x/10);
}
```

What is the output of the statement: `cout << F2(13579);`?

a) 9  
b) 1  
c) 135  
d) 579  
e) None of the above

31. What is the output of the program given in the two columns below?

```cpp
#include <iostream.h>
int a,b,c;

void confuse1(int b, int a)
{
    a += 1;
    b += 1;
    c += 1;
}

void confuse2(int b, int &a)
{
    a += 2;
    b += 2;
    c += 2;
}

void confuse3(int *b, int c)
{
    a += 3;
    *b += 3;
    c += 3;
}

void main()
{
    int a,c;
a=b=c=0;
    confuse1( a, b);
    confuse2( b, c);
    confuse3( &a, c);
    cout << a << b << c << endl;
}
```

a) 141  
b) 768  
c) 634  
d) 303  
e) None of the above
32. Consider the following section of code:

```c
int x, y, z;
int *xPtr, *yPtr, **zPtr;
x = 3; y = 7; z = 9;
xPtr = &x;
yPtr = &x;
zPtr = &xPtr;
```

The results of the statement

```
printf( "%d", **zPtr);
```

is:

a) 8  
b) 2  
c) 246  
d) 468  
e) None of the above

33. What is the output of the program given in the two columns below?

```c
#include <stdio.h>
#define MAX 5

void f3( int *intPointer)  
{  
int array[MAX] = {1,2,3,4,5};  
int i;  
for (i=0; i<MAX; i++)  
{  
    *intPointer = array[i];  
    intPointer++;  
}
}

void f4( int *intPointer)  
{  
    int i;  
    for (i=MAX-1; i>=0; i--)  
    {  
        printf("%d", *intPointer);  
        intPointer--;  
    }
}

main()  
{  
    int array[MAX];  
    int *index = array;  
    f3(index);  
    index = array + MAX - 1;  
    f4(index);  
    printf("\n");  
}
```

a) 12345  
b) 54321  
c) 01234  
d) 43210  
e) None of the above