You may take this test with you after the test, but you must turn in your answer sheet.

This test has the following sections:
   I. True/False .......................... 64 points; (32 questions, 2 points each)
   II. Multiple Choice................... 36 points; (6 questions, 6 points each)

100 points total

This test is worth 10% of your final grade. You must put your answers on the bubble form. 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. Be careful - more than one answer may seem to be correct. Some questions are tricky.

**True/False: (2 points each)** On your bubble form fill out A for true and B for false.

**T F 1.** Every C program can run in a C++ compiler.

**T F 2.** Every C++ program can run in a C compiler.

**T F 3.** The decimal number 45 in binary is 101101

**T F 4.** The binary number 1010101 equals the decimal number 83

**T F 5.** In C the strstr function can be used to concatenate strings

**T F 6.** Output of the following code segment is: True

```c
char word1[]="test";
char word2[]="test";
if( word1==word2) printf("True");
```

**T F 7.** Output of the following code segment is: 3

```c
int values[]={3,5,7,11};
printf(" %d", &values[ 0]);
```

**T F 8.** Output of the following code segment is: 3

```c
int values[]={3,5,7,11};
printf(" %d", values[ 0]);
```

**T F 9.** Output of the following code segment is: 65

```c
char letters[]="ABCD";
printf(" %d", (int)letters[ 0]);
```
10. Output of the following code segment is: A

```cpp
char letters[] = "ABCD";
cout << letters << endl;
```

11. Output of the following code segment is: A

```cpp
char letters[] = "ABCD";
cout << letters[0] << endl;
```

12. Output of the following code segment is: A

```cpp
char letters[] = "ABCD";
cout << &letters[0] << endl;
```

13. Output of the following code segment is: A

```cpp
char letters[] = "ABCD";
char *pLetters = &letters[0];
cout << pLetters << endl;
```

14. Output of the code segment shown below is: B

```cpp
char values[] = "ABCD";
char *pChar = values;
printf(" %c", pChar + 1);
```

15. Output of the code segment shown below is: B

```cpp
char values[] = "ABCD";
char *pChar = values;
printf(" %c", *pChar + 1);
```

16. Output of the code segment shown below is: B

```cpp
char values[] = "ABCD";
char *pChar = values;
printf(" %c", *(pChar + 1));
```

17. A char array can be passed to a function that catches it as a char *

18. A char * can be passed to a function that catches it as a char array
19. Output of the following code segment is: 1.5

```
int x = 3;
cout << x/2 << endl;
```

20. In a C program with an array declared using square brackets [], the size of the array can be a variable.

21. In a C program with an array declared using malloc, the size of the array can be a variable.

22. In a C program the grow command can be used to add storage to a one-dimensional array.

23. To pass a character array called word to function printIt we could use:

```
printIt( word[]);
```

24. If a 2D array of int were declared in the parameter list of a function, we would not need to specify the size of the first array dimension.

25. Assume the code in C++ shown below, where function swapValues is called.
Output of this segment of code is: Values are: 2 7

```
void swapValues(int *num1, int &num2)
{
    int temp = *num1;
    *num1 = num2;
    num2 = temp;
}

int x = 2;
int y = 7;
swapValues( &x, y);
cout << "Values are: " << x << " " << y << endl;
// ... other code
```

26. Assume the code in C shown below, where function swapValues2 is called.
Output of this segment of code is: Values are: 7 2

```
void swapValues2(int *num1, int *num2)
{
    int temp = *num1;
    *num1 = *num2;
    *num2 = temp;
}

int x = 2; int *pX = &x;
int y = 7; int *pY = &y;
swapValues2( pX, pY);
cout << "Values are: " << x << " " << y << endl;
// ... other code
```

27. To see if variable x is equal to or greater than variable y, we could use:

```
if( x => y) {
    printf("x => y");
}
```
T   F   28. The section of C++ code shown below would compile and run and give as output:

    Yes  Done

```
bool z = false;
if( z = true)
    printf("Yes ");
else
    printf("No ");
printf("Done \n");
```

T   F   29. The following code prints the words:  Hey there

```
char c='H';
switch (c){
    case 'H': printf("H");
    case 'e': printf("e");
    case 'y': printf("y");
    break;
}
printf(" there");
```

T   F   30. On average, selection sort is faster than bubble sort.

```
struct Node {
    int value;
    char name[ 15];
};
Node oneNode = {3,"Fred"};
Node *pNode = &oneNode;
```

T   F   31. The following statement would compile and run:

    cout << pNode->name << endl;

T   F   32. The following statement would compile and run:

    cout << *pNode.name << endl;
Multiple Choice (6 points each)

33. Consider the code segment shown below. If after the function call the value of number has changed, what is the most likely cause?

```c
int number = 5;
printf(" %d", number);
char letters[]="ABCD";
changeLetters( letters);
printf(" %d", number);
```

a) number is a global variable
b) Although number is not passed to function changeLetters, function changeLetters itself calls a second function which changes number
c) Function changeLetters overwrites the end of array letters
d) There is some ASCII control characters that are present in the code even though they are not visible
e) None of the above

34. When using a binary search, what is the maximum number of comparisons you would need to make when trying to find some number between 0 and 50,000?

a) 10
b) 15
c) 17
d) 19
e) None of the above

35. Consider the program shown at right below. What is the output?

```c
#include <iostream>
using namespace std;

char * doStuff( char words[])
{
    static char *pWords = words;
    char *pNext = strchr( pWords, ' ');
    *pNext = '\0';
    char * returnValue = pWords;
    pWords = pNext + 1;
    return returnValue;
}

int main()
{
    char words[]= "first second third ";
    cout << doStuff( words) << endl;
    cout << doStuff( words) << endl;
    cout << doStuff( words) << endl;
}
```

a) Three blank lines
b) f
c) first
second
d) It does not compile because of a pointer error
e) None of the above
36. What is the output of the following C++ program?

```cpp
#include <iostream>
using namespace std;

int x=3, y=5;

void confuse1(int y, int &x)
{
    x++;
    y++;
}

void confuse2(int *a, int b)
{
    y = ++b;
    *a = y++;
}

void confuse3(int &a, int *b)
{
    (*b)++;  
    confuse2( b, a);  
    a = *b;
}

int main()
{
    int x=1;
    confuse1( x, y);
    confuse2( &x, y);
    confuse3( x, &y);
    printf("x + y = %d \n", x+y);
    return 0;
}
```

(a) $x + y = 4$
(b) $x + y = 6$
(c) $x + y = 7$
(d) $x + y = 8$
(e) None of the above
37. Consider the two functions shown at right below. For positive numbers, how would you best describe the return value of calling function \textit{first}?

\begin{align*}
\text{a) } & x + y \\
\text{b) } & x \times x \\
\text{c) } & x \times y \\
\text{d) } & x \\
\text{e) } & \text{None of the above}
\end{align*}

\begin{verbatim}
int first( int x, int y)
{
    int z=1;
    for( int i=0; i<y; i++)
    { 
        z = second( z, x);
    }
    return z;
}

int second( int x, int y)
{
    int z = 0;
    for( int i=0; i<x; i++)
    {
        z += y;
    }
    return z;
}
\end{verbatim}

38. What is the output of the following C program?

\begin{verbatim}
#include <stdio.h>

int process( int *pNumber)
{
    int x=pNumber[0];
    int i;
    for( i=0; i<10; i++)
    { 
        if( *pNumber>x) 
        { 
            x=*pNumber;
        }
        pNumber++;
    }
    return x;
}

int main()
{
    int numbers[] = {3,1,5,12,7,3,9,11,15,8};
    int *pNumber = &numbers;

    printf(" %d\n", process( pNumber));
    return 0;
}
\end{verbatim}

\begin{align*}
\text{a) } & 1 \\
\text{b) } & 15 \\
\text{c) } & 3 \\
\text{d) } & 8 \\
\text{e) } & \text{None of the above}
\end{align*}