Open Ended Questions (26 pts):
1. (5 pts) What is the largest positive number and the smallest negative number that can be stored in a 16-bit, signed integer? What is the largest number that can be stored in a 16-bit, unsigned integer?

2. (5 pts) Convert the following binary to decimal (show your work!!!):
   a. 10
   b. 100
   c. 1010
   d. 11001
   e. 11010

3. (5 pts) Convert the following decimal to binary (show your work!!!):
   a. 3
   b. 8
   c. 10
   d. 22
   e. 30

4. (5 pts) Evaluate the following C expressions:
   a. 4 - 3 * 2 / 12
   b. 10 % 5 * 9
   c. 12.0 * 2 / 48
   d. 17 % 10 / 4.0
   e. (1 + 4) / (2 * 2)

5. (6 pts) Describe the difference between = and ==, and give examples of how to use each.
True/False and Multiple Choice (24 pts):

6. The standard convention in C is to name variables with lower case letters and name constants using all upper case letters. True False

7. Decomposition requires dividing a problem into smaller problems. True False

8. Every C program must contain a main function. True False

9. Creating functions requires you to take a general task and find a more specific category of task that it represents. True False

10. A literal value is the simplest expression. True False

11. C variable names can begin with a digit. True False

12. Curly braces are required around a single statement contained in a control structure. True False

(Only one possible answer for each multiple choice)

13. Which of the following is a valid variable name:
   a. $students
   b. Sstudents
   c. #students
   d. _students

14. What does the & in front of a variable name represent in C:
   a. contents of
   b. logical and
   c. address of
   d. inverse of

15. The declaration of a variable requires which of the following:
   a. name
   b. data type
   c. neither a nor b
   d. both a and b
16. The C compiler creates which of the following:
   a. a.out file
   b. file with the name supplied after -o
   c. neither a or b
   d. both a and b

17. Which of the following is used for commenting code in C?
   a. !
   b. #
   c. /* and */
   d. both a and c
   e. all of the above

Tracing/Debugging Code (50 pts):

18. (10 pts) Return statements change the flow of execution. It causes the program to “return” to the point where the function was called, and it can bring a value with it.
   • Describe and give examples of what this function can return. Your explanation should explain when each return statement is used.
   • How could you change this code, so that it only has a single return, at the end of the function? The code must have the exact same behavior.

```c
int what_do_i_return(int x){
    int i;

    if(x <= 1){
        return -2;
    }

    for(i = 1; i < 100; i++){
        if(i % 2 == 0 && i % 5 == 0 && i % x == 0){
            return i;
        }
    }

    return -1;
}
```
19. (20 pts) Identify/Circle and Explain at least 10 syntax/logic errors in the following code?

```c
void main() {
    int num==9;

    printf('Hello, world!\n');
    scanf("%d", num);    /*Read an integer from user
     message()

    return 0;

    void message {
        printf("This program surely cannot \n";
        printf("contain any "errors " in it!\n");
    }
```
20. (10 pts) What is the output from the following loop?

```c
int count, num = 4;
for (count = 1; count <= num; count++) {
    printf("%d\n", num);
    num /= 2;
}
```

21. (10 pts) What is the output from the following code?

```c
int num, total = 25;
for (num = 1; num <= (total / 2); num++) {
    total -= num;
    printf("%d %d \n", total, num);
}
```

Extra Credit (10 pts):
1. (5 pts) Convert the for loop in Question 20 to an equivalent while loop.

```c
int num;
printf("Enter a number: ");
scanf("%d", &num);
if(num % 2 == 0) {
    if(num % 3 == 0)
        printf("The number is divisible by 6!\n");
    else
        printf("The number is odd!\n");
}