Problem Set 1

Problem 1.1

(a) what do curly braces denote in C? Why does it make sense to use curly braces to surround the body of a function?

anw - Curly braces mean a block of code with the aim to complete one function - a function is a set of code to achieve one goal, curly brace makes it a whole block

(b) Describe the difference between the literal values 7, "7", and '7' anw

- 7 is a number, it could be int. "7" is a string, '7' is a char.

(c) Consider the statement

```
double ans = 10.0 + 2.0/3.0 - 2.0*2.0;
```

Rewrite this statement, inserting parentheses to ensure that ans = 11.0 upon evaluation of this statement.

anw

```
double ans = 10.0 + 2.0/ 3.0 - 2.0 *2.0 ;
```

Problem 1.2

Consider the statement

```
double ans = 18.0/squared(2+1);
```

For each of the four versions of the function macro squared() below, write the corresponding value of ans.

1. #define squared(x) x*x

ans: ans = 18.0/2+1*2+1 = 9.0+3 = 12.0

2. #define squared(x) (x*x)

ans: ans = 18.0/(2+1*2+1) = 18.0/5 = 3.6

3. #define squared(x) (x)*(x)

ans: ans = 18.0/(2+1)*(2+1) = 6.03 = 18.0

4. #define squared(x) ((x)*(x))

ans: ans = 18.0/((2+1)*(2+1)) = 18.0/9 = 2.0

Problem 1.3

Write the "Hello, 6.087 students" program described in lecture in your favorite text editor and compile and execute it. Turn in a printout or screen shot showing

- the command used to compile your program - the command used to execute your program (using gdb) - the output of your program

ans:

```
gcc 1_3.c -o 1_3.o
gdb 1_3.o
```

Problem 1.4

see 1_4.c

Problem 1.5

For each of the following statements, explain why it is not correct, and fix it.

(a) #include <stdio.h>;

ans: no need semicolon #include <stdio.h>

(b)

```
int function(void arg1)
{
    return arg1-1;
}
```

ans: the return type should be the same as input type

```
int function(int arg1)
{
    return arg1-1;
}
```

(c)

```
#define MESSAGE = "Happy new year!"
puts(MESSAGE);
```

ans:

```
#define MESSAGE "Happy new year!"
puts(MESSAGE);
```