Name: Enter your name here.

**Problem 1 – 12 pts.**

What does the following code print?

#include <stdio.h>

int foo(int a)

{

a = a + 1;

return a;

}

int main()

{

int a = 1;

int b = 5;

int c = foo(foo(b));

int d = foo(foo(foo(b))) + foo(foo(a)) + 10;

printf("a = %d\n", a);

printf("b = %d\n", b);

printf("c = %d\n", c);

printf("d = %d\n", d);

return 0;

}

OUTPUT:

a = 1

b = 5

c = 7

d = 21

**Problem 2 – 10 pts**

What does the following code print?

#include <stdio.h>

int foo(int a, int b)

{

if (b == 0)

{

return 1;

}

return a \* foo(a, b-1);

}

int main()

{

int a = 3;

int b = 5;

int c = foo(b, a);

printf("c = %d\n", c);

return 0;

}

OUTPUT

c = 125

**Problem 3 – 12 pts.**

What does the following code print?

#include <stdio.h>

int foo(int a, int b);

int bar(int c, int d);

int main()

{

int a = 3;

int b = 5;

int c = foo(b, a);

printf("c = %d\n", c);

return 0;

}

int foo(int a, int b)

{

if (b <= 1)

{

return 1;

}

return a \* bar(a, b-2); 5\*

}

int bar(int c, int d)

{

if (c <= 1)

{

return 1;

}

return d + foo(d, c);

}

c = 10

**Problem 4 –**

I have provided a file called problem4.c that contains an incomplete program. The goal of the program is to get a word from the user and print out the word so that each letter is printed multiple times. Complete the program by defining two functions:

1. A function called repeatLetters that satisfies the following specs:
   1. repeatLetters takes two arguments, called text and times.
   2. The function prints each letter of text as many times as specified by times.
2. A function called getTimes that satisfies the following specs:
   1. Asks the user for an integer greater than 0
   2. If the user enters an integer less than 1, prints an error message and keeps asking them until they enter a valid integer. See the example output for the format.
   3. Returns the valid integer.

I have written main. It is your job to write the functions. You can add them directly to the source file I provide. You are not allowed to modify main in any way.

This is an example run of the program:

Enter some text, or q to quit: hello

Enter number of times (must be > 0): 3

hhheeellllllooo

Enter some text, or q to quit: good morning

Enter number of times (must be > 0): 2

ggoooodd mmoorrnniinngg

Enter some text, or q to quit: a b

Enter number of times (must be > 0): 7

aaaaaaa bbbbbbb

Enter a word or q to quit: cabbage

Enter number of times (must be > 0): 0

0 is <= 0, try again.

Enter number of times (must be > 0): 1

cabbage

Enter some text, or q to quit: q

**Problem 5 – pts**

I have provided a file called problem5.c that contains an incomplete program. The goal of the program is to ask the user for a radius and calculate the volume of a sphere with that radius. Complete the program, without changing main, by writing the following three functions :

1. a function called getRadius that satisfies the following specs:
   1. The function asks for the user to enter a number. Get the number as a float. See the example output for the format.
   2. If the user enters a number other than -1 or something >= 0, print an error message and ask for a number again. See the example output for the format.
   3. Otherwise, return the number the user entered.
2. a function called cubeFloat that satisfies the following specs:
   1. Takes a float called number as an argument.
   2. To avoid using the math.h library and having to manually link the library, the function should simply return number\*number\*number.
3. a function called sphereVolume that:
   1. Takes a radius as input
   2. Applies the formula for calculating the volume of a sphere found below, using your cubeFloat function to cube the radius. You can use the PI that I created with #define.
   3. Returns the volume.

Here is an example run of the program, so you can get the formatting correct:

Enter a radius >= 0 or -1 to quit: -23

The number should be >= -1.

Enter a radius >= 0 or -1 to quit: 1

Volume = 4.19

Enter a radius >= 0 or -1 to quit: 2.3

Volume = 50.96

Enter a radius >= 0 or -1 to quit: 4.6

Volume = 407.72

Enter a radius >= 0 or -1 to quit: -1

If your volume results are not the same as mine, check whether you are using the appropriate type of division (floating point vs. integer).

**Problem 6 – 18 pts**

I have provided a file called problem6.c that contains an incomplete program. The goal of the program is to determine which character in a string occurs most often and how many times it occurs. Complete the program by writing the following two functions:

1. Function countOccurences takes two arguments, as string and a character. It returns the number of times the character occurs in the string.
2. Function mostFrequentCharacter takes a string as an argument. It returns the character that occurs most often in the string. If multiple characters tie for occurring the most times, the function can return any one of the tied characters. This function should make use of your countOccurrences function.

Uppercase and lowercase characters should not be considered the same, so ‘A’ and ‘a’ are different characters.

**Hint**: This is a maximization problem, as you’re trying to find the character that occurs the maximum number of times. It is also a counting/accumulation problem, because you need to count how many times something occurs. For the counting part, you create a sum = 0 and each time you come across the character you’re looking for in the string, you increment sum. For the maximization part, initialize a variable max\_char to store the character that occurs the most times. You can assign the first character in the string to it, as that will be the first character that occurs more than 0 times and will thus be your first max character. Also initialize a variable max\_counter to keep track of the maximum number of times a character occurs. For each letter in the string, count how many times it occurs. If it occurs more times than max\_counter, then update both max\_counter and max\_char to reflect the new character and count.

Here is an example run of the program, so you can get the formatting correct:

Enter a word or q to quit: hello

Most frequent character: l

Number of occurrences of l: 2

Enter a word or q to quit: 788788878901

Most frequent character: 8

Number of occurrences of 8: 6

Enter a word or q to quit: Q

**Submission:**

Place this Word document and the three program source files (problem4.c, problem5.c, problem6.c) in a folder and zip the folder. Submit the zip file in Canvas. Make sure that the zip file contains your source code and Word document before submitting.