Label the various parts of this function (body, name, etc):

int sumOfRange(int a, int b) {

int sum = 0;

for(int i = a; i <= b; i++)

sum += i;

return sum;

}

What does this code do? And how is it used (called) in a program?

It sums all values from a to b, and returns that sum. It is called with “sumOfRange(a,b)” and returns an int value.

Write a function called “sum\_from\_to” that takes two integer parameters (begin, and end) and returns the sum of values from begin to end. Invert the numbers if begin is greater than end. You function should handle both positive and negative values.

For example:

cout << sum\_from\_to(4, 5) << endl; // would print 9

cout << sum\_from\_to(0, 3) << endl; // would print 6.

cout << sum\_from\_to(3, 0) << endl; // would also print 6

Write a function called “***enough***” that takes one integer parameter (goal) and returns an integer result. The result is determined by adding consecutive values starting at 1 in a loop until you reach (but do not exceed) the goal. The last value added will be returned as the result.

Examples:

For a goal of 9:

1 + 2 + 3 = 6 so the function would return a 3

1 + 2 + 3 + 4 = 10 which is greater than 9 so the return value has to be 3

For a goal of 10:

1 + 2 + 3 + 4 = 10 so it would return a 4

For a goal of 2:

1 + 2 =3 is larger than two so it would return a one.

int enough(int goal){

int sum = 1

int i = 2;

for (i = 2; sum < goal; i++)

{

sum += i;

}

return i – 1;

}

This function will implement the present value formula given below:

P = F / ( 1 + r)n

The terms of the formula are as follows:

* P is the *present value*, or the amount of money that you need to deposit today.
* F is the *future value* that you want in the account.
* r is the *annual interest rate* (expressed in decimal form)
* n is the *number of years* that you plan to let the money set in the account

Write a function called **presentValue** that performs this calculation. It takes as parameters *future value, annual interest rate*, and *number of years*. It should return the *present value*, which is the amount you need to deposit today.

Here is the function prototype to get you started:

double presentValue(double F, double r, double n);

In this function you will need to determine how many pizzas to purchase based on the following:

* Number of people
* Number of slices each person can eat
* How large each pizza to be purchased as a ***diameter*** in inches.

The function should take all the above as parameters and return the number of pizzas needed to feed the party goers. Make sure you round up the number of pizza’s to the next whole pizza.

***Please use a descriptive function name, proper data types for the parameters, and proper function return type to get full credit.***

**Problem solving hints:**

A surface area of slice of a single of pizza is 14.125 square inches.

The surface area area of pizza can found using the formula:

Pizza Area = 3.1415927 X pizza radius X pizza radius

Here is an example:

Diameter is 12-inches

Number of people is 4

Number of slices per person is 2

Pizza area = 3.1415927 X 6 X 6 = 113.0 square inches

Number of slices per pizza = 113/14.125 = 8

Number of slices of pizza needed = 4 X 2 = 8

So we need only one pizza for four people. Not much pizza but hey we are cheap!

Write a function called “***arbitraryTriangleArea”*** that returns a bool and takes 5 parameters of which as follows:

side\_a side\_b side\_c ***t\_area t\_perimeter***

The last two are ***pass-by-reference*** variables used to return the area of the triangle and the perimeter.

You will need to use appropriate data types.

Your function should check to see if the triangle is valid and if it is, calculate the area, the perimeter and return true. If the triangle is not valid, return false and set the values of t\_area and t\_perimeter to -1. ***We are not necessarily dealing with a right triangle, so you will need to check the sum of any pair of any two sides is larger than the third.***

**Formula**:

Area =

where , and , , and are each of the sides of an arbitrary triangle.