GNG 1106 Lab 2 report

Pythagorean triples

Step 1: Problem identification and statement:

Calculate the number of possible Pythagorean triples (a,b,c) for a positive integer (N) with c<N. Also, calculate the Possible Pythagorean triples for that positive integer (N) and the triple with the largest value of c.

Step 2: Gathering of information and Input and Output description:

A Pythagorean triple is a set of three numbers (integers) a,b and c ($0 < a < b < c$*). This Pythagorean triple follows the equation* $a^2 + b^2 = c^2$*. A well-known example is the triple (3,4,5). This triple simply represents the sides of a right triangle where the integer 5 is the hypotenuse of the triangle. Nevertheless, non-integer sides do not form a Pythagorean triple. For example, (*$1,1,\sqrt{2}$*). In this lab, we wish to compute the amount of Pythagorean triples which respect c <N. This simply means that we wish to obtain every triple that has a hypotenuse value smaller than the integer that was inputted. We would also wish to compute the number of Pythagorean triples possible and the triple with the largest value of c. In order to compute all the required information, we need to understand the basic and fundamental logic behind these triples, starting with the smallest possible triple (3,4,5). Here is an example of how to calculate (manually) all possible Pythagorean triples using 3, 4 and 5 as a reference:*

Let "n" be any integer greater than 1, then 3n, 4n and 5n are also a set of Pythagorean Triple. This is true because:

$$(3n)^2 + (4n)^2 = (5n)^2$$

Examples:

n	**(3n, 4n, 5n)**
2	(6,8,10)
3	(9,12,15)
...	... etc ...

So we can make **infinitely many** triples just using the (3,4,5) triple.

Although, there are some triples that can be found by trial and error using the Pythagorean triple. (see example below, #3)

Let's illustrate this, using an example:

What are the possible Pythagorean triples for 15?

1. *(3,4,5)*
2. *(3(2),4(2),5(2)) = (6,8,10)*
3. *Let's create a triangle for this triple : (5, 12, 13)*

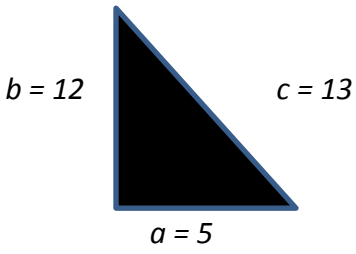
$a^2+b^2=c^2$

$(5)^2 + (12)^2 = c^2$

$169 = c^2$

$13 = c$

Thus, (5,12,13) is a Pythagorean triple which respects c<15.



This is the triangle for the calculated Pythagorean triple.

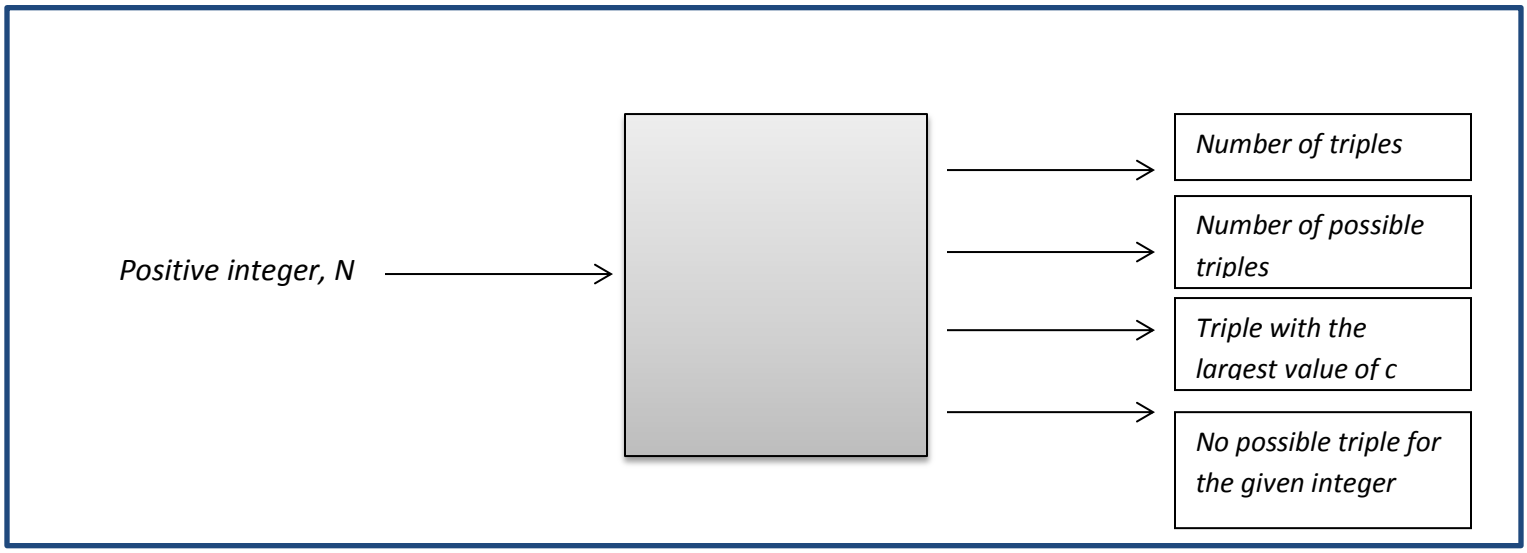
We conclude by saying that there are 3 possible Pythagorean triples in that range and the three are as follows :

1. *(3,4,5)*
2. *(6,8,10)*
3. *(5,12,13)*

And the largest possible triple is (5,12,13)

There is an easier way to do all of these calculations using the C code and it is described in my algorithm in the following step of my lab report.

The I/O diagram for this problem is illustrated below.



Thus, the <u>inputs</u> are: The positive integer (represented by variable N)

The <u>outputs</u> are: the number of triples, the number of possible triples, the triple with the largest value of c and no possible triple for the given integer.

Step 3: Test Cases and Algorithm Design

Test Cases:

The following table provides 4 test cases that can be used to test the algorithm and software during the lab. Here are a different set of positive integer values and their respective triplets (respecting c<N)

Test Case	***Positive integer (N)***	***Number of triples***	***Pythagorean triples (a,b,c)***	***Largest Pythagorean triple (a,b,c)***
1	*5*	*0*	*There is no Pythagorean triple in this range*	*N/A*
2	*15*	*3*	*(3, 4, 5)* *(6, 8, 10)* *(5, 12, 13)*	*(5, 12, 13)*
3	*24*	*6*	*(3, 4, 5)* *(6, 8, 10)* *(5, 12, 13)* *(9, 12 ,15)* *(8, 15, 17)* *(12, 16, 20)*	*(12, 16, 20)*
4	*26*	8	*(3, 4, 5)* *(6, 8, 10)* *(5, 12, 13)* *(9, 12 ,15)* *(8, 15, 17)* *(12, 16, 20)* *(15, 20, 25)* *(7, 24, 25)*	*(7, 24, 25)*

Algorithm:

Input a value
Input b value
Input c value
Input max 1, max2, max3
Input N value
Input whichNumber
Input count value

Print "Please enter a positive integer"
Read a value into N (positive integer)
If N is smaller or equivalent to 5, print "There is no Pythagorean triple in this range"
Else,
For "c" equivalent to zero, smaller than N, increment c
For "b" equivalent to zero, smaller than c, increment b

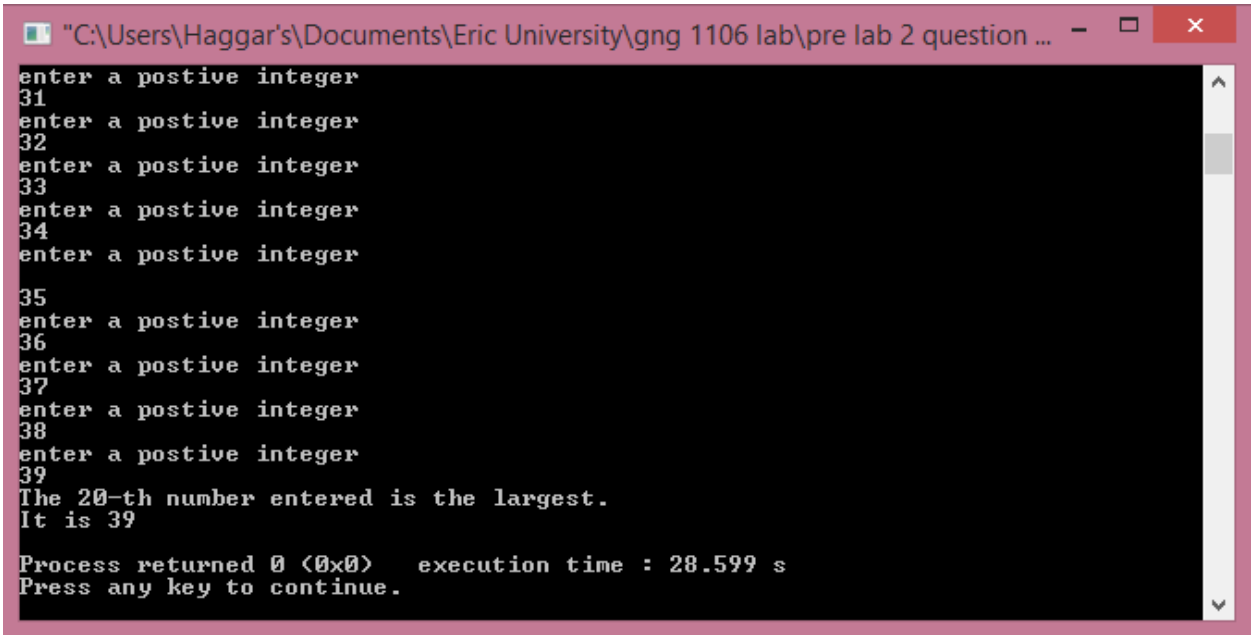
For "a" equivalent to zero, smaller than b, increment a
*If (a*a + b*b = c*c)*
Print the triples

Set Max1 equivalent to "a"
Set Max2 equivalent to "b"
Set Max3 equivalent to "c"
Set whichNumber equivalent to count increased by 1
Print "the triple that has the largest value of c"
Print "the number of possible Pythagorean triples"

Additional Pre-Lab Preparation

1. *This program asks the user to input 20 positive integers. After the 20th integer inputted, the program will output the largest value entered as well as the placement of this value in the 20 integer serie.*

   For example, if we enter the all numbers from 20 to 39(exactly 20 numbers), the output of the program is as follows (screenshot of the program) :



   Thus, we obtain the placement of the largest value in the serie and its value.

2. ***<u>The following block of code will print :</u>***

0 0; 1; 2; 3; 4; 5;

1 1; 2; 3; 4; 5;

2 2; 3; 4; 5;

3 3; 4; 5;

4 4; 5;

5 5;

6

7

Step 4: Implementation

The C program developed to solve my problem is listed below.

```
/*-------------------------------------------------*/
 /* Name: Eric Haggar, Student Number: 7674509 */
 /* Date: October 9, 2014. */
 /* Program: calculating Pythagorean triples */
 /* Description: This program computes the number of Pythagorean
triples, the Pythagorean triples and the Pythagorean triple with
the largest value of c*/
 /* using a positive integer. */
/*-------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
void main ()

{

/*The following integers are assigned to the main function and
will be used throughout the code */


  int a = 0, b = 0, c = 0, max1, max2, max3, N, whichNumber;

  int count = 0;


/*First off, the user will be asked to enter a positive integer.
Once entered, the program will assign the integer to N*/

  printf("Please enter a positive integer: \n");

  scanf("%d", &N);

  if (N <=5){

    printf("There is no Pythagorean triple in this range.\n");

  }

  /* if N is smaller than 5, the program will print : There is
no Pythagorean triple in this range*/

    else{

            /* Create a repeat sequence so that the program
computes every possible value for a,b and c within the given
range

            and that follows the requirement 0<a<b<c<N */

  for (c = 0; c < N; c++)

  {

      for (b = 0; b < c; b++)

      {

          for (a = 0; a < b; a++){
```

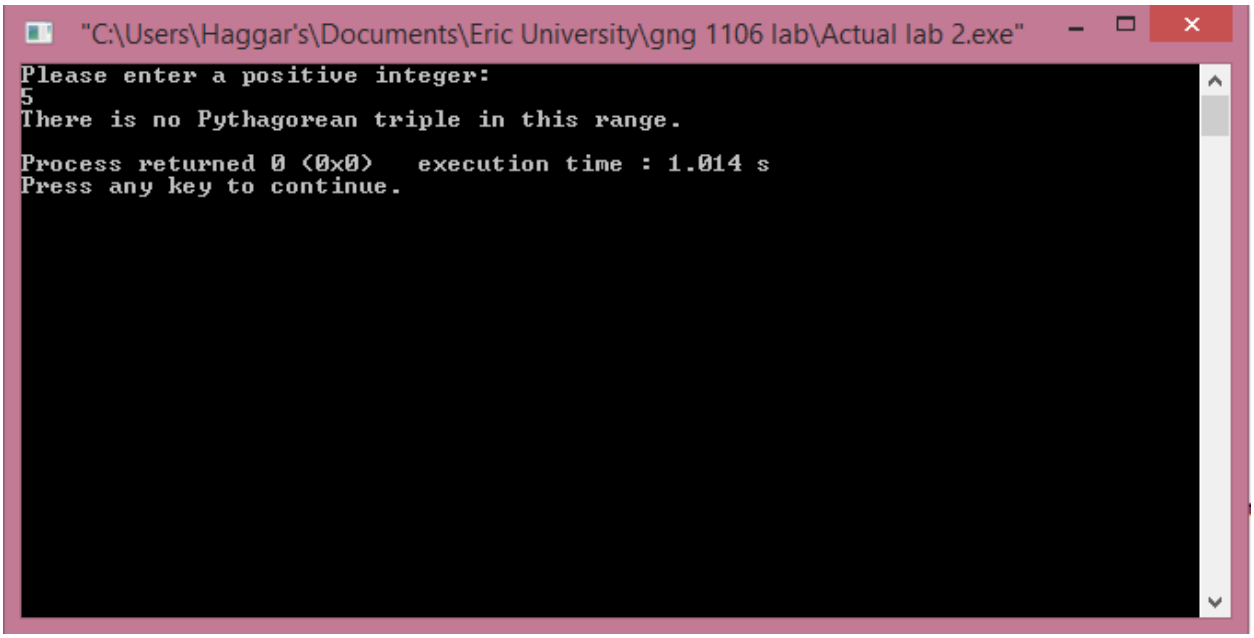
```
            if (a * a + b * b == c * c ){

                  /* Each for loop will be printed if the
equation a * a + b * b == c * c is respected*/

                printf("%d: \t%d %d %d\n", ++count, a, b, c);

                max1 = a;

                max2 = b;

                max3 = c;

                whichNumber = count +1;

              }

          }

      }

  }

/* in order to obtain the triple that has the largest value of
c, we assign max 1 to a, max 2 to b and max 3 to c*/

/* The output will be the respective triple for the largest
value of c*/

printf("The triple that has the largest value of c is %d %d
%d\n", max1,max2,max3);

printf("The number of possible Pythagorean triples are %d\n",
count);


    }

}


/*---------------------------End----------------------*/
```

Step 5: Tests and Verification (and Debugging)

<u>Test Case 1:</u> For positive integer 5, the output is:

There is no Pythagorean triple in this range

Screenshot of the program:



which is in agreement with the test case expected output.

<u>Test Case 2:</u> For positive integer 15, the output is:

1: (3, 4, 5)

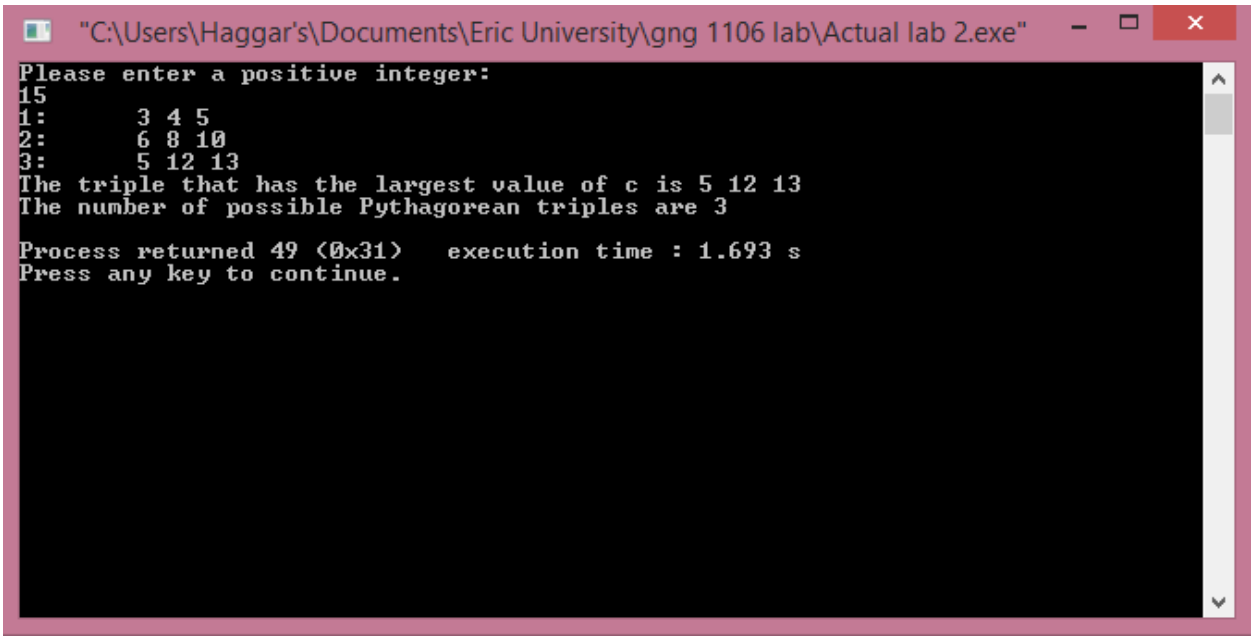
2: (6, 8 , 10)

3: (5, 12, 13)

The triple that has the largest value of c is : 5, 12, 13

The number of possible Pythagorean triples are : 3

Screenshot of the program:



which is in agreement with the test case expected output.

<u>Test Case 3:</u> For positive integer 24, the output is:

1: (3, 4, 5)

2: (6, 8, 10)

3: (5, 12, 13)

4: (9, 12, 15)

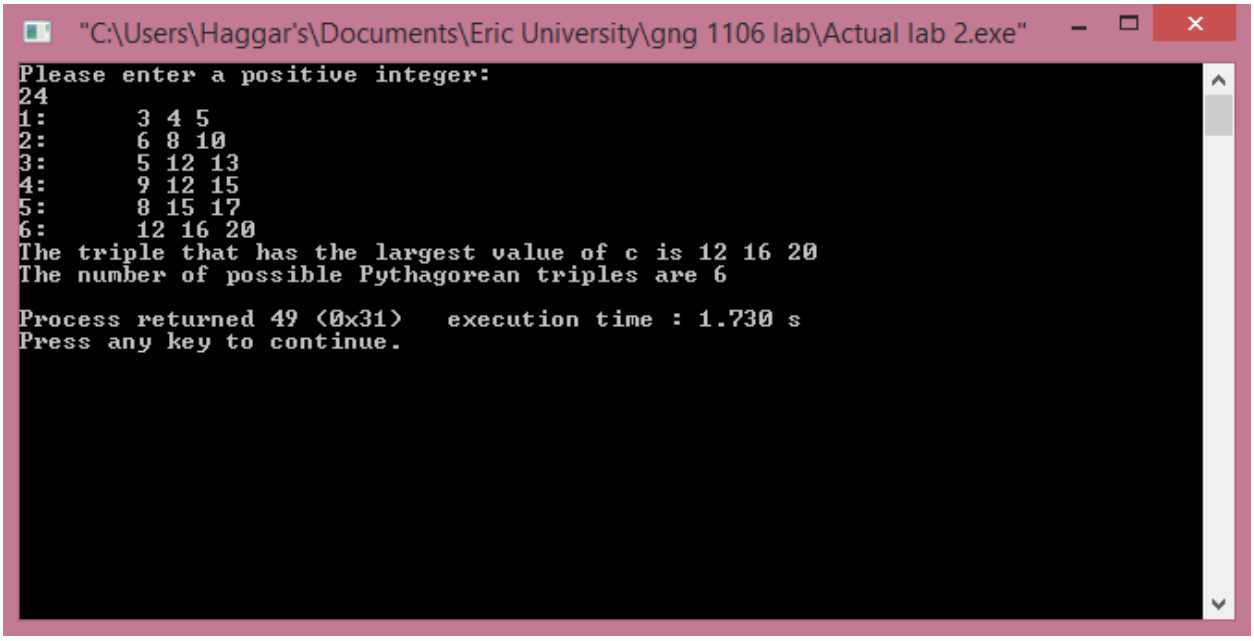
5: (8, 15, 17)

6: (12, 16, 20)

The triple that has the largest value of c is : 12, 16, 20

The number of possible Pythagorean triples are : 6

Screenshot of the program:



which is in agreement with the test case expected output.

Test Case 4: For positive integer 26, the output is:

1: (3, 4, 5)

2: (6, 8, 10)

3: (5, 12, 13)

4: (9, 12, 15)

5: (8, 15, 17)

6: (12, 16, 20)

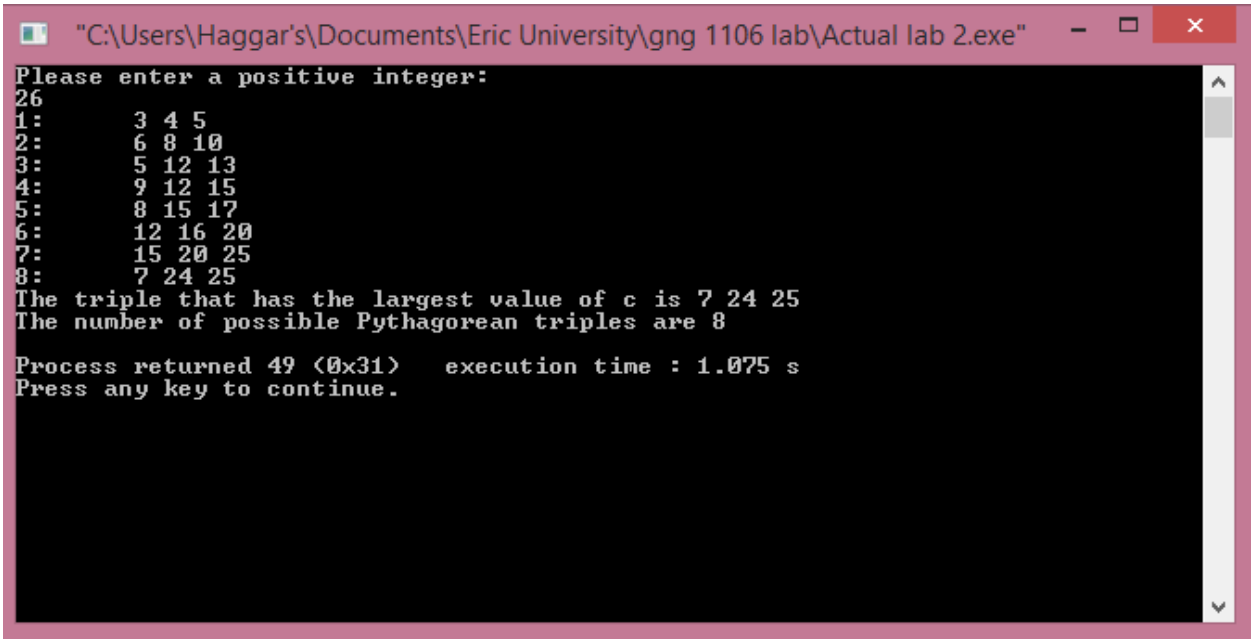
7: (15, 20, 25)

8: (7, 24, 25)

The triple that has the largest value of c is : 7, 24, 25

The number of possible Pythagorean triples are : 8

Screenshot of the program:



which is in agreement with the test case expected output.

We can conclude that the program is functioning correctly.

Conclusion:

In conclusion, during the lab, I encountered one problem that led to the failure of one element of the program. I wasn't able to print the Pythagorean triple with the largest value of c. Every time I would print, it would give me 3 numbers, that didn't form the expected Pythagorean triple. I simply assigned max 1 to a, max 2 to b and max 3 to c and added them in my "printf" statement and the right Pythagorean triple was outputted. Therefore, the program executes successfully and all the information required is outputted by inputting one positive integer N.