UNIVERSITY GARDENS HIGH SCHOOL

SAN JUAN, PUERTO RICO

Pythagorean Triples in the Pascal Triangle:

A computational and algebraic approach

By

Jan-Paul V. Ramos

ABSTRACT

The Pascal Triangle isn't necessarily useful when it comes to other types of triangles, but here we find that it can be used as a tool to find a set of Pythagorean Triples. This shows the usefulness of the Pascal Triangle as a mathematical tool, thus giving it more purpose. The second diagonal line in the Pascal Triangle is composed by the triangular series. This sequence can be used to find Pythagorean Triples, for the known reason that when any b in a Pythagorean Triple is divided by four, you get a number from the sequence, which is also a fraction of a data tree of primitive triples. To find {a,b,c}, each variable needs to be able to take n in terms of a chosen row in the Pascal Triangle where the n-integer is, making full use of the Pascal Triangle. A new equation expresses the set in such a way that you can find any Pythagorean Triple by replacing any n with the previously mentioned row. This new set must be proven algebraically with the Pythagorean Theorem. Software is introduced to create a user- friendly application to understand mathematical findings, and help mathematicians verify their work through proof assistant programming languages. The proof assistant programming language is quintessential to this process, as it is used to verify the previously mentioned algebraic proof computationally. This method for making use of the Pascal Triangle is important for the average person to understand derivation of Pythagorean Triples and to motivate professionals to take a computational approach in their everyday workload.

‘

TABLE OF CONTENTS

Introduction

The Pascal Triangle is an array of binomial coefficients who's purpose lies on the person's interest. With such an interesting arrangement of numbers, many mathematicians have been able to find patterns such as the Fibonacci sequence, and the Triangular Number Series, in addition to many more. Although these patterns are interesting, not many people have found a use for them, as they just represent how most sequences are related to a certain pattern of binomial coefficients, the actual use of the Pascal Triangle in these number theory problems is not particularly used. This investigation focuses on finding a useful purpose for the Pascal Triangle, in which students can learn through number theory and perceive a certain subject in a different view. This paper focuses on the relationship between the Pascal Triangle and Pythagorean Triples (a set of numbers defined as *a, b, c* that satisfy the Pythagorean Theorem), in addition to computational methods to prove this relation.

By utilizing knowledge of how the Pascal Triangle is formed, this investigation formulates a new equation that provides a certain set of Pythagorean Triples while only utilizing the Pascal Triangle, more specifically, the rows in the Pascal Triangle. This is all possible thanks to the composition of such triangle, which includes a certain set of numbers called the Triangular Numbers. These Triangular Numbers are well known for being useful in finding a certain set of Pythagorean Triples, this set being such that for all *b* in any Pythagorean Triple, this *b* has to be divisible by 4 and output any element in the set of natural numbers. This provides us with all the possible Pythagorean Triples that can be derived from the Pascal Triangle utilizing our new formula.

With this useful formula, it is now possible to derive any Pythagorean Triple that complies with the previous restrictions. Such formula can be used for making connections regarding the Tree of Primitive Pythagorean Triples, a new way of generating such data tree, as evidenced by Price's tree of primitive Pythagorean Triples [cite here], they meet the restrictions, which can provide another way of

writing algorithms utilizing the Pascal Triangle. Furthermore in the methodology, the investigation focuses on programming a simple user-interface (written in C++) that explains how the formula works, and gives the user an opportunity for them to derive their own Pythagorean Triple while using the Pascal Triangle. This gives people who don't know anything about number theory a brief introduction into the subject and can help find such Pythagorean Triples that satisfy a specific restriction a lot easier, an example being Primitive Pythagorean Triples, a triple in which all elements in the set do not have any common divisor. Another important part of computation during this method is the use of a proof assistant, which as the same suggests, verifies a mathematical proof computationally. This tool is showcased to emphasize the importance of computing in mathematics and to prove that mathematicians can be more efficient and secure by learning how to program in these proof assistants.

While analyzing the results, a formula is proven mathematically and computationally using the Coq proof assitant program, thus proving that the formula actually works for all cases. A comparison of a Primitive Pythagorean Tree and the possible Pythagorean Triples generated from the formula is statistically made and shows and obvious connection between the Pythagorean Triples generated from the Pascal Triangle and a data tree, thus giving the Pascal Triangle a wider purpose while generating Pythagorean Triples.

Justification

In the simplest terms a Pascal Triangle is a triangular array of the binomial coefficients. The importance of such a triangle is useful in finding patterns in subjects such and number theory, combinatorics, and even algebra. Although the Pascal Triangle has been studied for centuries by various mathematicians from very different backgrounds (Boyer, 2018), today it is not used as a main tool of finding a mathematical answer. For example, we could utilize such triangle to calculate the coefficients for $(x+y)^n$ where n is the row in the Pascal Triangle. Even with this useful trick, there are other ways we teach students and mathematicians use to derive such answers because of either efficiency or personal preference, preferring to actually derive the formula to find such coefficients or just thorough multiplication can be more understandable and easier to remember for students, as the average person and professional doesn't have an infinite Pascal Triangle in their pocket to calculate any coefficient whenever they want. By developing software that does the process of finding the n^{th} row for someone, and calculating the output, while still showing them the triangle and explaining the process, this can end up being a useful tool for the average person, but it still doesn't necessarily make a professionals job easier, a mathematician to be exact. While the average person just makes use of the results, a mathematician wants to derive and prove the equation rigorously to make sure it works with every possible input. In this case, a proof assistant programming language can be used to verify the mathematician's work, and would be helpful to remove any middleman that would have to verify the work. With these software applications, it is clear how important the combination of mathematics and technology is in modern day proofs and learning. This can motivate mathematicians to use programming frequently in their work and increase in efficiency.

In this investigation the application of software is used specifically in the use for the Pascal Triangle as a tool for finding Pythagorean Triples. This also leads to other uses in Pythagorean Triples,

one of those being a data tree of Primitive Pythagorean Triples, in which a new way of writing algorithms can be derived from the Pascal Triangle. Furthermore, a Pythagorean Triple is simply a set of three numbers *{a, b ,c}* that satisfy the Pythagorean Theorem (The Editors of Enyclopaedia Britannica, 2018). It is common knowledge that in the Pascal Triangle we can find the Triangular Series in the second diagonal (reading from left to right and starting with the first diagonal at number 0). The Triangular Series can be expressed with the explicit formula **(Appendix A)**. There are many ways to prove this, but the main purpose of the series is that it counts objects arranged in an equilateral triangle, making a sequence of the next number being a sum of the previous ones plus the current term (The Editors of Enyclopaedia Britannica, 2019). By using the previously mentioned series, an intuitive and interesting formula can be derived to find Pythagorean Triples utilizing just the Pascal Triangle. This can be used as a tool to find Pythagorean Triples in a more efficient way while dealing with number theory problems, which would mainly be seen in an academic setting.

The formula for finding each Pythagorean Triple may be intuitive itself, albeit by utilizing software this investigation uses it as an educational tool as well. With a small program with a basic user-interface, anyone can view the Pascal Triangle and choose any n^{th} row to find the Pythagorean Triple correspondent to the number in the Triangular Series. This can be useful for anyone who is not familiar with the Pascal Triangle or Pythagorean Triples themselves, in addition, provides them with an intuitive way of finding these Pythagorean Triples, even without the triangle in front of them. A more complex problem that can be solved with help of software is proving a mathematical concept. With help of computational proof assistants, a mathematician can learn to program their proof and have a computer verify almost anything and make sure their proof has no errors.

Literary Revision

I. Mathematical Foundation

The Pascal Triangle is defined as a triangular array of numbers (Hosch, 2013). This provides a method of finding different solutions to problems in combinatorics, number theory, and even algebra. The most common example for the use of the triangle is to find the coefficients of $(x+y)^n$, where *n* is the row in the Pascal Triangle that is made up of the coefficients, in that same order (from left to right) (Hosch, 2013). It is also important to note that these rows start counting from the number 0, instead of our convential counting system starting from 1. Although the previously mentioned triangle was named after French mathematician Blaise Pascal, who is also known as one of the creators of the first mechanical calculators, there has been evidence of the triangle being used in civilizations way before Pascal's time in 1654 (Jerphagnon & Orcibal, 2020). This triangle has served as a convinient trick to find the outcome of situations like coin tosses, the amount of coin tosses can represent the row in the Pascal Triangle which has the possible outcomes in it’s coefficients (. From finding prime numbers to the Fibonacci sequence, the Pascal Triangle certainly is a gold mine of combinatorics, number theory, and mathematics in general.

Apart from Blaise Pascal, there is another famous mathematician from ancient times, who contributed so much to the field that his theorem’s are still used to this day and have been studied rigorously by mathematicians throughout the millenia. This person is known as Pythagoras, who was not just a mathematician, but also a philosopher, whom founded the Pythagorean brotherhood (Encyclopaedia Britannica, 2013). Pythagoras contributed philosophical thoughts and thus from him generated Pythagoreanism, which displays confusing and complex ideas in philosophy, such as metaphysics, and easier subjects like music. Aside from the arts, closely related to Pascal’s investigation, Pythagoreanism studied numbers and mathematics in ways that could benefit engineering

methods, the most famous one being the Pythagorean Theorem. The Pythagorean Theorem is a well-known world wide as mainly a geometric theorem. This theorem suggests that the sum of the legs squared in a triangle will equal the length of it's hypotenuse (Encyclopaedia Britannica, 2018). Much like Pascal's Triangle, even though the concept is attributed to the more modern mathematician, the theorem has been found centuries before Pythagoras' time, dating from dates as far back as 1750 B.C.E in Babylonian tablets (Encyclopaedia Britannica, 2018). What was found specifically was not the theorem itself, it was the set of possible integers that satisfy the equation, known as Pythagorean Triples.

Pythagorean Triples are well known in geometry because of how useful they are in finding the legs and hypotenuse in a right angled triangle. Triples can be found in many ways, but the most studied once outside of geometry are primitive Pythagorean Triples. A primitive Pythagorean Triple consists of a set of numbers that are coprime, meaning that no more than the number one is a common factor of both numbers. Not necessarily do all numbers have to be prime, but they have to satisfy for being coprime. By using elementary number theory we can find any primitive Pythagorean Triple in which the numbers inputted will be relatively prime integers and neither can be odd (Ribenboim, 1984). Another way for finding Pythagorean Triples is by using the Triangular Series numbers. There is a specific set of Pythagorean Triples where every *b* in the subsets are the Triangular Series numbers divided by four, using this knowledge anyone could derive the set for the Pythagorean Triple.

The Triangular Series is interesting, because as the name suggests, it can be found in a triangle like shape, it can even be found in a Pascal Triangle. There are explicit formulas used to derive an amount of numbers in the Triangular Series, which can be observed as the same formula that can be used to find the sum of n-integers, in which the next number in the sequence is the sum of all the previous numbers plus the current term. The series itself is just made up of an unknown variable to form an equilateral triangle (Sloane, 2008). Such a thing can be proved in many ways, some of them

being by either using a visual proof, which is just proving by what you can visually determine because of how self-explanatory it is. Another type of proof is proof by induction. This is used mainly when someone wants to prove for every possible integer, so they prove for x and the successor of x, which would end up being written as $x+1$ (Encyclopedia Britannica, 2018). Although it is commonly used, the logic behind it is argued by mathematicians. Some include mathematical induction as one of the five axioms for arithmetic, showing that there is no difference from this and other postulates in arithmetic in other mathematical branches.

II. Programming Foundation

As mathematics grows in complexity through time, so does our society. From ancient times to just this past turn of the millennium, mathematics was only done by mathematicians with a pencil and paper. Today we have developed engineering and software methods that make understanding mathematics and developing it in the workplace a lot easier. This is through the creation of programming languages. A programming language can be explained as a set of rules that a user writes to tell a computer to compute your instructions (Hemmendinger, 2019). Originally this was done through mathematical algorithms in giant computers that ran using bits and you could not really program, but instead make inputs for the computer to receive through electrical processes in the transistors, the place where each bit is stored. Nowadays people have developed more readable and executable programming languages which store the written commands into the bits for the user.

One of the most important programming languages for when it comes to mathematical development are algorithmic languages. These are designed for the specific reason to express mathematical computations, which can either be algebraic operations or any type of mathematics. Out of all of these, the most common ones include FORTRAN, ALGOL, LISP, and C (Hemmendinger, 2019). Most of these have been replaced with higher end and more modern languages such as the replacement of C for C++, which is just an addition to the C programming language, and the Python

programming language, which is one of the fastest growing programming languages in the world for it's use in absolutely anything (Heath, 2017). There are also other types of programming languages, such as business-oriented languages, used for data and systems in a company, education-oriented languages, which includes such languages as BASIC and Pascal (inspired by famous mathematician Blaise Pascal), Logo, and Hypertalk. The there can be found object-oriented languages, which are the most used ones today besides algorithmic ones, and they include C++, Ada, Java, and Visual BASIC and some examples (Hemmendinger, 2019). There are declarative languages, which are higher level that focus on what to do and are usually iterative, and these include functional languages as a subset, another commonly used language set, which include Haskell, ML, LISP, and more. Finally there are scripting languages which are usually used for creating operating systems (Hemmendinger, 2019).

Through time, programming languages have developed to suit the needs of programmers and engineers throughout time. Recently however, there have been developments by mathematicians in the theoretical side of the spectrum that have created useful languages for mathematical processes. A famous example of this is MATLAB, which is used for numerical analysis (Atkinson, 2017). Another big one is Coq, which works as an interactive theorem prover, in which a mathematician can verify their proof by programming it. These languages are useful for the development of mathematics as a modern outlook is given instead of the classical pen and paper (Saxton, 2017).

Problem

Finding a formula to derive Pythagorean Triples from the Pascal Triangle, and any n^{th} term from the Triangular Series, and use computational methods to proof it and demonstrate computation can be used for proofs in mathematics. With these results, find a correlation between this newly found set of Pythagorean Triples and a data tree which can be newly generalized using just the Pascal Triangle and the formula. After the fact, create a user-interface in which the results can be shown in an educational way to those unfamiliar with the Pascal Triangle, and show a computational proof made in a proof assistant.

Hypothesis

By using simple algebra a formula to find Pythagorean Triples in the Pascal Triangle can be found, in addition to a new way of getting any number in the Triangular Series. All possible Pythagorean Triples will form fraction of a Primitive Pythagorean Triple Tree, thus finding a new way of generating said tree, or better yet, a “branch”, or fraction, of it.

Methodology

I. Mathematics

1. Generating Formula

By noticing the Triangular Series in the Pascal Triangle, a known formula for finding Pythagorean Triples in the Pascal Triangle is noted: *{a, b , c} => {2n-3, 4m, 4m+1},* where *b* is a multiple of 4. The variable n is any number from the sequence, while m is the row in the Pascal Triangle from which the n was retrieved. It is important to note that in this formula the Pascal Triangle's rows start at 1. To rewrite the formula in terms of only using the Pascal Triangle, the binomial coefficient **(Appendix B)** that can be used to arrange a Pascal Triange , where *n* determines the row and *b* determines the elements in that row. Such notation is expressed as **(Appendix C)**. For programming purposes later on and to follow the Pascal Triangle's rules, the second element in the row, starting now from the first element being the 0^{th} term, the second term will always be a number of the Triangular series. This notation is usually reffered to as *n choose b*, which in this case would be *n choose 2.* **(Appendix D)**

2. Proof

By simplifying the equation, an equation is given in which by choosing any n^{th} row in the Pascal Triangle, starting with the first row at 0, the output will always be a number in the Triangular Series. This means that the *m* in the original equation can be replaced with this new equation, which would have to be multiplied by 4 for *b* and the result added 1 for *c*, *a* staying the same. The new Pythagorean Triple formula would be: *{2(n + 1) - 3, 2n(n - 1), 2n(n - 1) + 1}*, in which *n* is in terms of the row of the Pascal Triangle. An algebraic proof is made by plugging in *{a, b , c}* in the Pythagorean Theorem to make sure it satisfies all possible values. **(Appendix E)**

Thus proven algebraically, the set is done. It is important to note that, done by inspection of the first couple sets of Pythagorean Triples: *{3, 4, 5} | {5, 12, 13} | {7, 24, 25}*, ever set is composed of a primitive Pythagorean Triple in which *c* is one greater than *b*, thus proving the existence of all possible sets generated from this formula to be in a "branch" in a primitive Pythagorean Triple Tree. **(Appendix F)**

II. Programming

1. C++

Using Visual Studio Code as a text editor, and compiling the program in Bash 4.4.20 terminal, a C++ program is made to show a demo of how these Pythagorean Triples are generated. **(Appendix G)**, **(Appendix H)**, **(Appendix I)**, **(Appendix J)**

2. Coq

For time and deadline purposes, the only conducted proof on Coq was the proof to calculate any number in the sum of n-integers, which just so happens to be the set of numbers in the Triangular Series. This program proofs the sum of n-integers set and calculates any n^{th} term in the sequence in the range of 1-322. The code uses a proof by induction method. **(Appendix K)**, **(Appendix L)**

Data Analysis

Both the algebraic proof by hand and the Coq proof turned out with no errors and proved the mathematical concepts, and the C++ program compiled with no problem, serving the goal of teaching the found formula. A Triangular Series n^{th} number formula is also retrieved from the Pascal Triangle. The set of Pythagorean Triples generated by the formula is a set that can be found in a branch of a primitive Pythagorean Triple tree. This is certified by the fact the top "branch", or a certain fraction, in such data tree will always be composed of Pythagorean Triples with the same restrictions as the ones that can be found with the formula. **(Appendix M)**

Conclusion

The investigation succeeded in most of the thought of predictions. The hypothesis was proven right and sets of Pythagorean Triples could be derived directly utilizing the Pascal Triangle. This data provides us with not just a new way of getting Pythagorean Triples, albeit a formula to generate any n^{th} element from the Triangular Series. The generated sets also satisfied a fraction or a "branch" of a primitive Pythagorean Triple tree, which allows to formulate this fraction with the Pascal Triangle. A program in C++ was successfully finished and is a simple user-interface which anyone can access through the GitHub repository and learn how this formula works and generate any Pythagorean Triple they want, with a certain limitation on how big the Pascal Triangle can get in the program. This provides anyone with not just the knowledge of the formula, but also the Coq proof is in the repository, which provides mathematicians with an explained example of how to make a proof computationally.

Some limitations were the maximum amount of lines the Pascal Triangle prints at in the user-interface, a maximum of 24, because of formatting errors in the program. Another problem encountered was while creating the algebraic proof in Coq, which proved to be too time consuming and could not be finished before the deadline, while an unfinished version remains in the repository.

Projections

With the newly found data, there is a possibility of generalizing a fraction of a primitive Pythagorean Triple tree utilizing the Pascal Triangle, which would lead to utilizing the Triangular Series and binomial theorem to define such fraction. Another fact is that the GitHub repository can be updated with a finished version of the Pythagorean Triple proof in Coq, along with a computational proof for the Triangular Series with the found formula. Finally, the user-interface created in C++ can be upgraded into a fully usable website which can be more accessible to anyone without having to need any programming knowledge to compile such code, and make it into an actual tool non-mathematicians and students can use to find Pythagorean Triples and learn number theory subjects regarding the Pascal Triangle.

Acknowledgments

I am grateful for the teachers at University Gardens High School for giving me the opportunity to do this investigation, more specifically thankful towards Prof. Xavier Pagán and Prof. José Cruz from said school for helping me formulate and organize my investigation and the paper. I am also helpful to those who helped verify the calculations, Prof. Marcel Ruiz and Carlos A. Alvarado Álvarez.

References

Boyer, C. B. (1950). Cardan and the Pascal triangle. *The American Mathematical Monthly*, *57*(6), 387- 390.

Hemmendinger, D. (2019, October 10). Object-oriented programming. Retrieved from https://www.britannica.com/technology/object-oriented-programming

Hosch, W. L. (2013, February 24). Pascal's triangle. Retrieved from https://www.britannica.com/science/Pascals-triangle

Jerphagnon, L., & Orcibal, J. (2020, January 8). Blaise Pascal. Retrieved from https://www.britannica.com/biography/Blaise-Pascal

Ribenboim, P. (2012). *The book of prime number records*. Springer Science & Business Media.

Saxton, D. (2011, February). An Introduction to Computer Programming and Mathematics. Retrieved from https://nrich.maths.org/6873

Sloane, N. J. (2004, February 19). A000217. Retrieved from https://oeis.org/A000217

The Editors of Encyclopaedia Britannica. (2018, March 1). Pythagorean theorem. Retrieved from https://www.britannica.com/science/Pythagorean-theorem

The Editors of Encyclopaedia Britannica. (2019, January 18). Binomial theorem. Retrieved from https://www.britannica.com/science/binomial-theorem

The Editors of Encyclopaedia Britannica. (2018, April 17). Mathematical induction. Retrieved from https://www.britannica.com/science/mathematical-induction

Thesleff, H. (2013, February 21). Pythagoreanism. Retrieved from https://www.britannica.com/science/Pythagoreanism

Appendix

Appendix A Triangular Series Formula for the Triangular Series

$$T_n = \sum_{k=1}^{n} k = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} = \binom{n+1}{2}$$

Appendix B Binomial Coefficient Notation for binomial coefficient

$$\binom{n}{k}$$

Appendix C Binomial Coefficient Formula Expression of the binomial coefficient mathematically

$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

Appendix D Choose 2 Math operation to find the second element in the Pascal Triangle (new formula)

$$Choose\,2 \equiv \binom{n}{2} \equiv \frac{(n!)}{(((n-2)!)2!)} = \frac{(n(n-1)(\cancel{n-2})(\cancel{n-3}))}{(2(\cancel{n-2})(\cancel{n-3})\ldots)} = \frac{(n(n-1))}{2}$$

Appendix E Algebraic Proof Proof by plug-in to verify the newly found Pythagorean Triple set works for every possible viable input

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

$$(2n(n-1)+1)^2=(2n(n-1))^2+(2(n+1)-3)^2$$

$$4n^2(n+1)^2+4n(n-1)+1=4n^2(n-1)^2+4(n-1)^2-12(n+1)+9$$

$$4n(n-1)+1=4(n+1)^2-12(n+1)+9$$

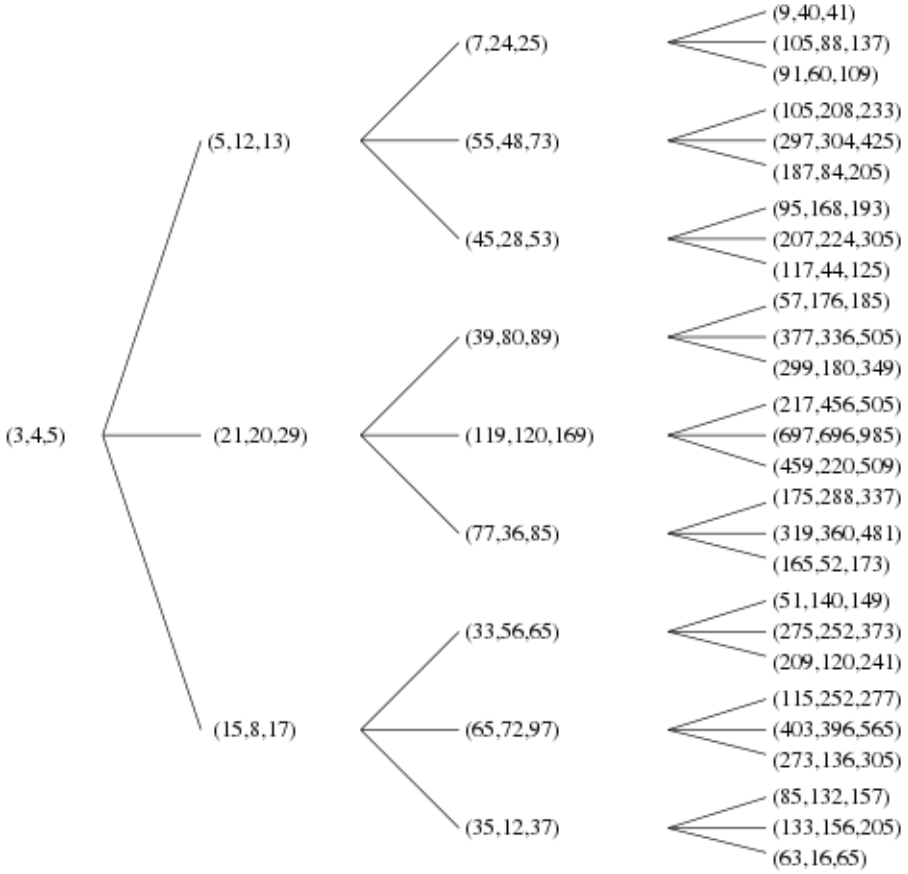
$$4n^2-4n+1=4n^2+8n+4-12n-12+9$$

$$-4n+1=(8n-12n)+(4-12+9)$$

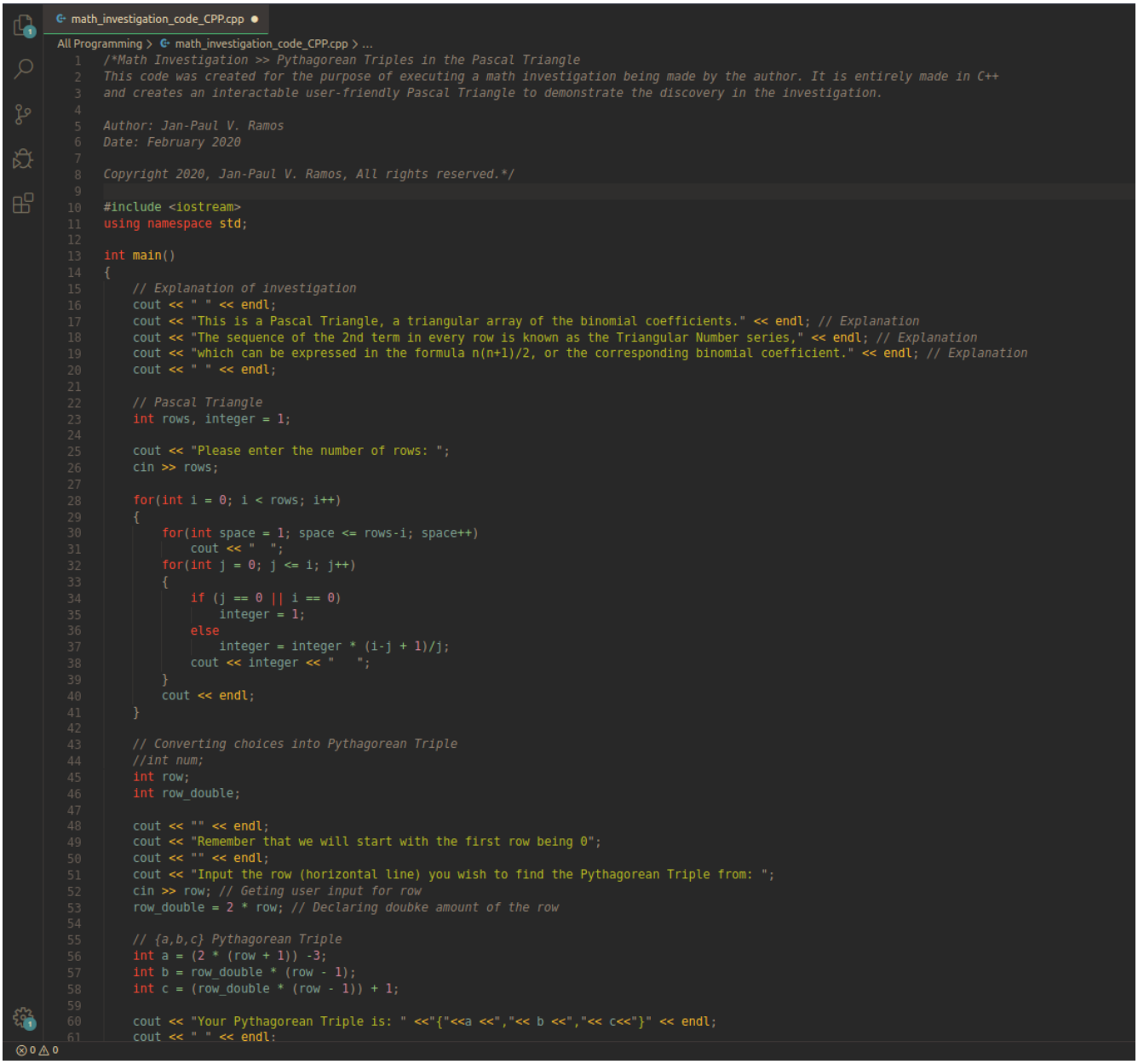
$$-4n+1=-4n+1$$

$$0=0$$

Appendix F Primitive Pythagorean Triple Tree Berggrens's tree of primitive Pythagorean triples



Appendix G C++ 1 First part of the C++ program



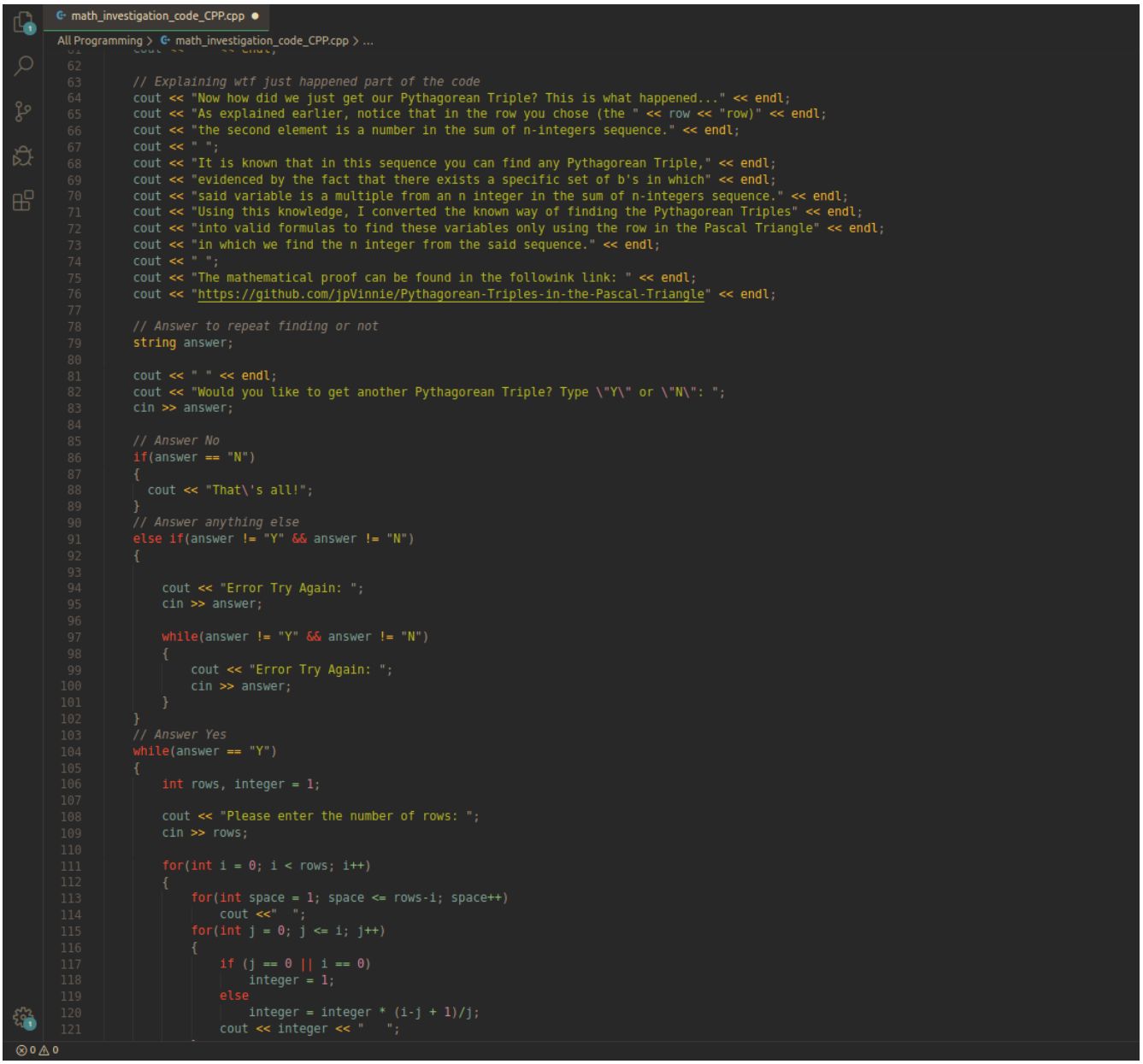
math_investigation_code_CPP.cpp

All Programming > math_investigation_code_CPP.cpp > ...

```cpp
/*Math Investigation >> Pythagorean Triples in the Pascal Triangle
This code was created for the purpose of executing a math investigation being made by the author. It is entirely made in C++
and creates an interactable user-friendly Pascal Triangle to demonstrate the discovery in the investigation.

Author: Jan-Paul V. Ramos
Date: February 2020

Copyright 2020, Jan-Paul V. Ramos, All rights reserved.*/

#include <iostream>
using namespace std;

int main()
{
    // Explanation of investigation
    cout << " " << endl;
    cout << "This is a Pascal Triangle, a triangular array of the binomial coefficients." << endl; // Explanation
    cout << "The sequence of the 2nd term in every row is known as the Triangular Number series," << endl; // Explanation
    cout << "which can be expressed in the formula n(n+1)/2, or the corresponding binomial coefficient." << endl; // Explanation
    cout << " " << endl;

    // Pascal Triangle
    int rows, integer = 1;

    cout << "Please enter the number of rows: ";
    cin >> rows;

    for(int i = 0; i < rows; i++)
    {
        for(int space = 1; space <= rows-i; space++)
            cout << "  ";
        for(int j = 0; j <= i; j++)
        {
            if (j == 0 || i == 0)
                integer = 1;
            else
                integer = integer * (i-j + 1)/j;
            cout << integer << "    ";
        }
        cout << endl;
    }

    // Converting choices into Pythagorean Triple
    //int num;
    int row;
    int row_double;

    cout << "" << endl;
    cout << "Remember that we will start with the first row being 0";
    cout << "" << endl;
    cout << "Input the row (horizontal line) you wish to find the Pythagorean Triple from: ";
    cin >> row; // Geting user input for row
    row_double = 2 * row; // Declaring doubke amount of the row

    // {a,b,c} Pythagorean Triple
    int a = (2 * (row + 1)) -3;
    int b = row_double * (row - 1);
    int c = (row_double * (row - 1)) + 1;

    cout << "Your Pythagorean Triple is: " <<"{"<<a <<","<< b <<","<< c<<"}" << endl;
    cout << " " << endl;
```

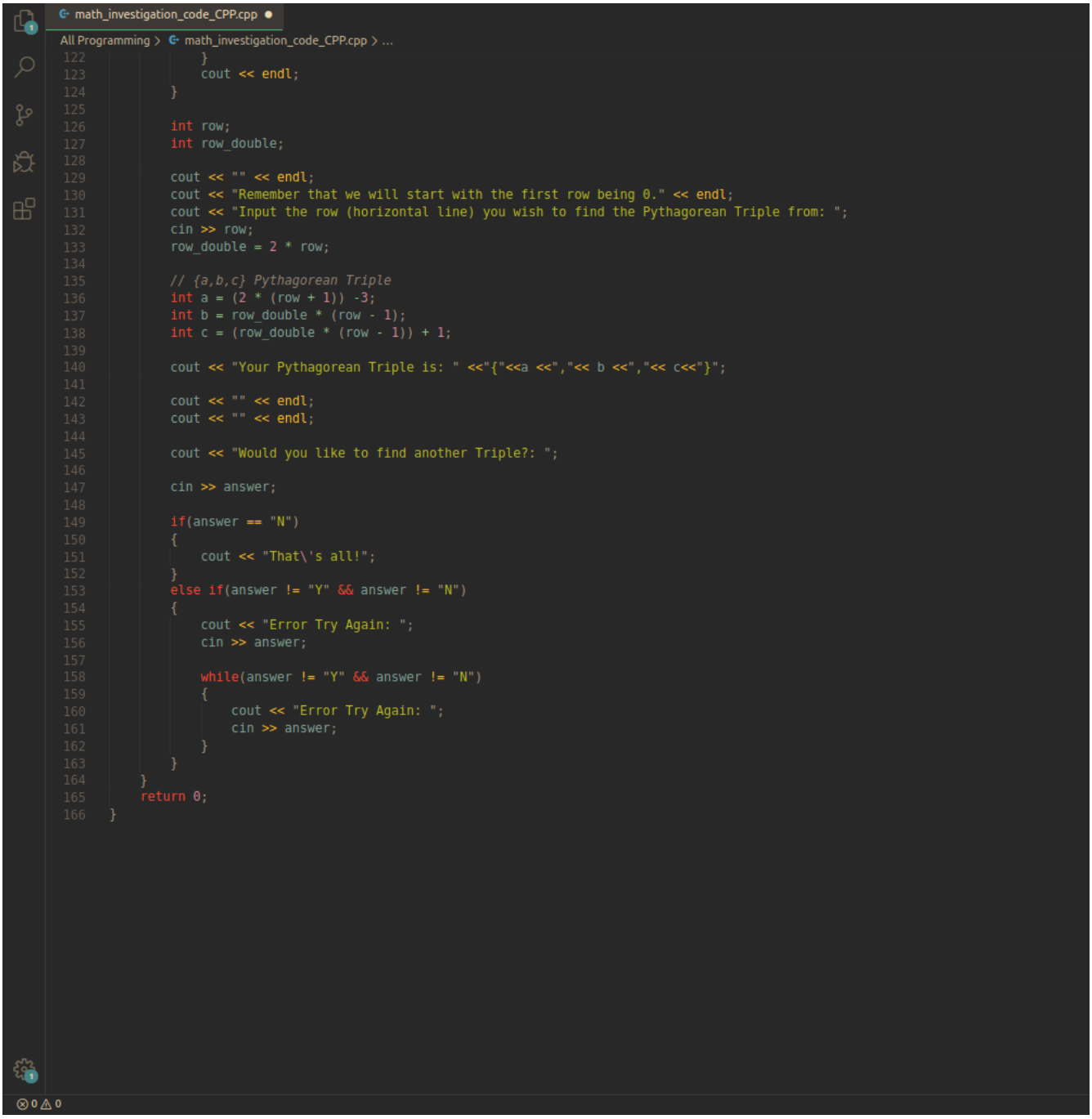
⊗0 ⚠0

Appendix H C++ 2 Second part of the C++ program




```
62
63      // Explaining wtf just happened part of the code
64      cout << "Now how did we just get our Pythagorean Triple? This is what happened..." << endl;
65      cout << "As explained earlier, notice that in the row you chose (the " << row << "row)" << endl;
66      cout << "the second element is a number in the sum of n-integers sequence." << endl;
67      cout << " ";
68      cout << "It is known that in this sequence you can find any Pythagorean Triple," << endl;
69      cout << "evidenced by the fact that there exists a specific set of b's in which" << endl;
70      cout << "said variable is a multiple from an n integer in the sum of n-integers sequence." << endl;
71      cout << "Using this knowledge, I converted the known way of finding the Pythagorean Triples" << endl;
72      cout << "into valid formulas to find these variables only using the row in the Pascal Triangle" << endl;
73      cout << "in which we find the n integer from the said sequence." << endl;
74      cout << " ";
75      cout << "The mathematical proof can be found in the followink link: " << endl;
76      cout << "https://github.com/jpVinnie/Pythagorean-Triples-in-the-Pascal-Triangle" << endl;
77
78      // Answer to repeat finding or not
79      string answer;
80
81      cout << " " << endl;
82      cout << "Would you like to get another Pythagorean Triple? Type \"Y\" or \"N\": ";
83      cin >> answer;
84
85      // Answer No
86      if(answer == "N")
87      {
88        cout << "That\'s all!";
89      }
90      // Answer anything else
91      else if(answer != "Y" && answer != "N")
92      {
93
94          cout << "Error Try Again: ";
95          cin >> answer;
96
97          while(answer != "Y" && answer != "N")
98          {
99              cout << "Error Try Again: ";
100             cin >> answer;
101         }
102     }
103     // Answer Yes
104     while(answer == "Y")
105     {
106         int rows, integer = 1;
107
108         cout << "Please enter the number of rows: ";
109         cin >> rows;
110
111         for(int i = 0; i < rows; i++)
112         {
113             for(int space = 1; space <= rows-i; space++)
114                 cout <<"  ";
115             for(int j = 0; j <= i; j++)
116             {
117                 if (j == 0 || i == 0)
118                     integer = 1;
119                 else
120                     integer = integer * (i-j + 1)/j;
121                 cout << integer << "    ";
```

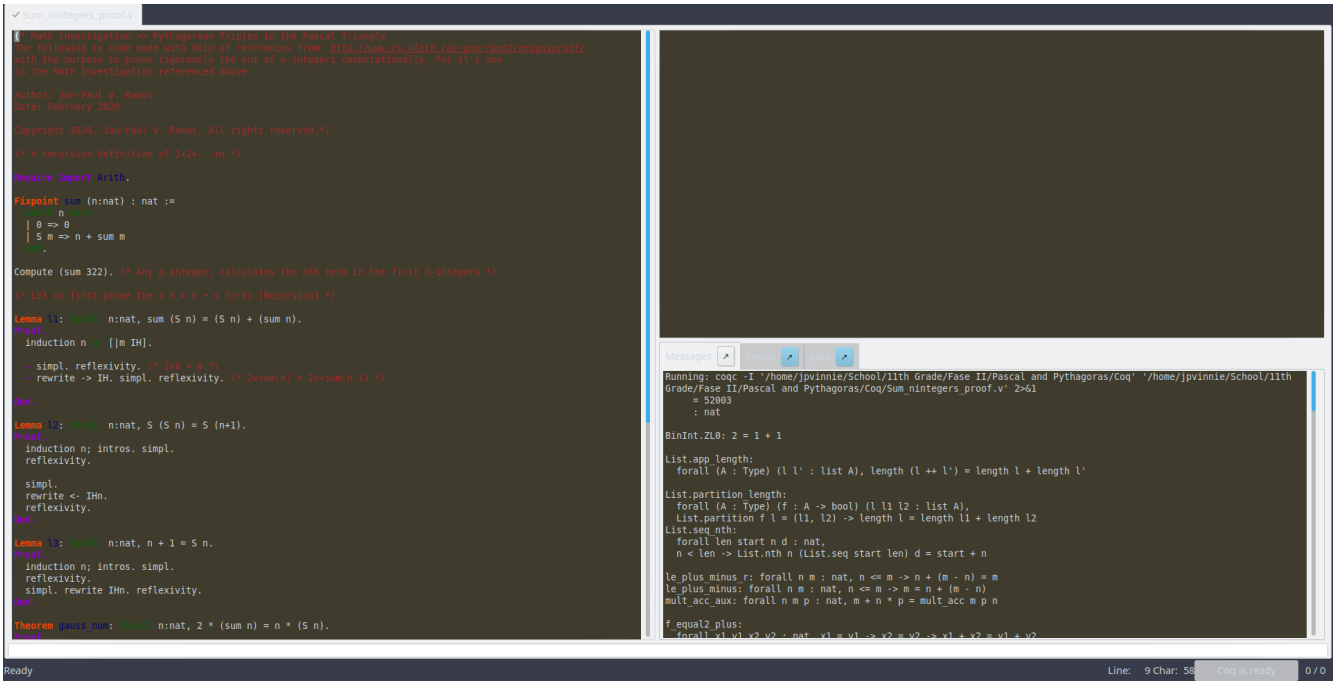
Appendix I C++ 3 Final part of the C++ program



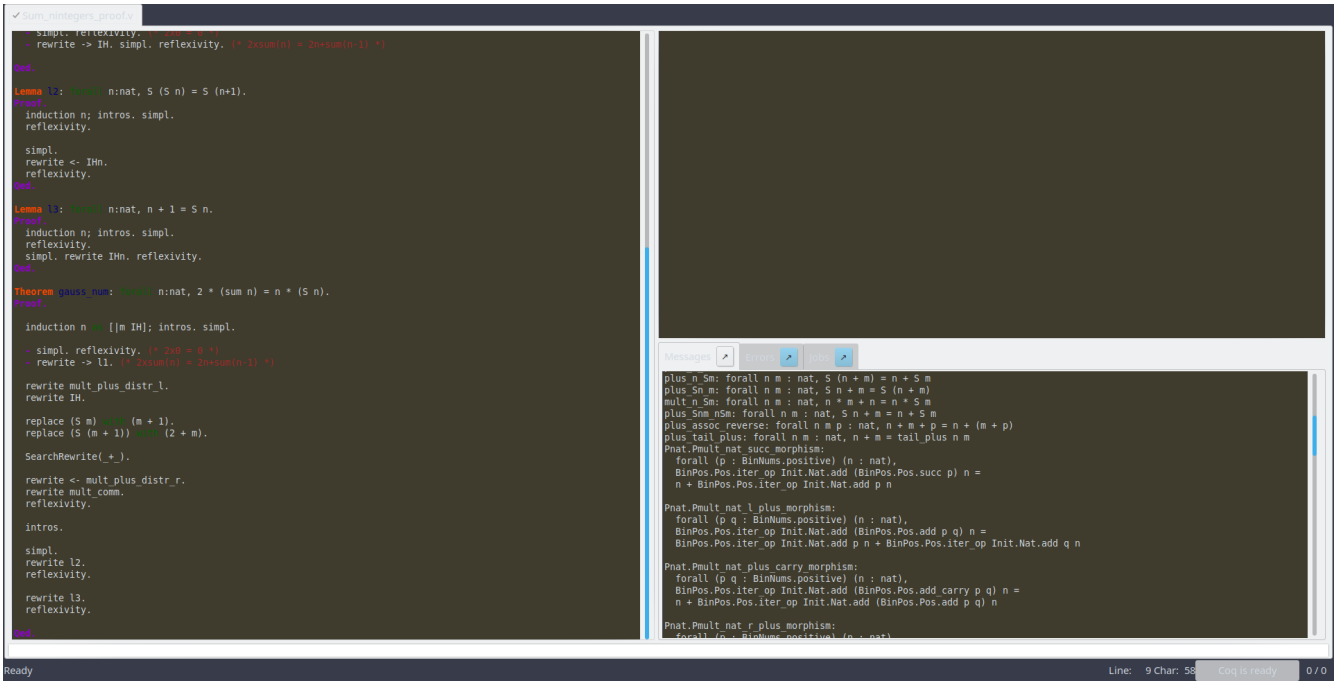
Appendix J Bash Terminal Compilation of the C++ code and running in bash terminal

```
[jpvinnie@stacys-mom]> ~/School/11th Grade/Fase II/Pascal and Pythagoras/All Programming$ ./Official_Program

This is a Pascal Triangle, a triangular array of the binomial coefficients.
The sequence of the 2nd term in every row is known as the Triangular Number series,
which can be expressed in the formula n(n+1)/2, or the corresponding binomial coefficient.

Please enter the number of rows: 10
                    1
                  1   1
                1   2   1
              1   3   3   1
            1   4   6   4   1
          1   5   10   10   5   1
        1   6   15   20   15   6   1
      1   7   21   35   35   21   7   1
    1   8   28   56   70   56   28   8   1
  1   9   36   84   126   126   84   36   9   1

Remember that we will start with the first row being 0
Input the row (horizontal line) you wish to find the Pythagorean Triple from: 3
Your Pythagorean Triple is: {5,12,13}

Now how did we just get our Pythagorean Triple? This is what happened...
As explained earlier, notice that in the row you chose (the 3row)
the second element is a number in the sum of n-integers sequence.
 It is known that in this sequence you can find any Pythagorean Triple,
evidenced by the fact that there exists a specific set of b's in which
said variable is a multiple from an n integer in the sum of n-integers sequence.
Using this knowledge, I converted the known way of finding the Pythagorean Triples
into valid formulas to find these variables only using the row in the Pascal Triangle
in which we find the n integer from the said sequence.
 The mathematical proof can be found in the followink link:
https://github.com/jpVinnie/Pythagorean-Triples-in-the-Pascal-Triangle

Would you like to get another Pythagorean Triple? Type "Y" or "N": Y
Please enter the number of rows: 15
                          1
                        1   1
                      1   2   1
                    1   3   3   1
                  1   4   6   4   1
                1   5   10   10   5   1
              1   6   15   20   15   6   1
            1   7   21   35   35   21   7   1
          1   8   28   56   70   56   28   8   1
        1   9   36   84   126   126   84   36   9   1
      1   10   45   120   210   252   210   120   45   10   1
    1   11   55   165   330   462   462   330   165   55   11   1
  1   12   66   220   495   792   924   792   495   220   66   12   1
1   13   78   286   715   1287   1716   1716   1287   715   286   78   13   1
1   14   91   364   1001   2002   3003   3432   3003   2002   1001   364   91   14   1

Remember that we will start with the first row being 0.
Input the row (horizontal line) you wish to find the Pythagorean Triple from: 8
Your Pythagorean Triple is: {15,112,113}

Would you like to find another Triple?: N
That's all![jpvinnie@stacys-mom]> ~/School/11th Grade/Fase II/Pascal and Pythagoras/All Programming$
```

Appendix K Coq 1 First part of the Coq proof of the sum of n-integers



Appendix L Coq 2 Second part of the Coq proof of the sum of n-integers



Appendix M PT Comparison Comparison between generated Pythagorean Triples and primitive Pythagorean Triples that satisfy the same restrictions

PT Comparison

Comparison between generated Pythagorean Triples and primitive Pythagorean Triples

Generated PT	Primitive PT from tree (top fraction)
{3,4,5}	{3,4,5}
{5, 12, 13}	{5, 12, 13}
{7, 24, 25}	{7, 24, 25}
{9, 40, 41}	{9, 40, 41}
{11, 60, 61}	{11, 60, 61}
{13, 48, 85}	{13, 48, 85}
{15, 112, 113}	{15, 112, 113}
{17, 144, 145}	{17, 144, 145}
{19, 180, 181}	{19, 180, 181}
{21, 220, 221}	{21, 220, 221}
{23, 264, 265}	{23, 264, 265}
{25, 312, 313}	{25, 312, 313}
{27, 364, 365}	{27, 364, 365}
{29, 420, 421}	{29, 420, 421}
{31, 480, 481}	{31, 480, 481}
{33, 544, 545}	{33, 544, 545}
.	.
.	.
.	.
{a, b, c}	{a, b, c}

*These primitive Pythagorean Triples are just a fraction, or a “branch”, of a full data tree.