# Triangular Numbers

## Introduction

A triangular number is a number of objects that can be arranged into an equilateral triangle like you see in the images below. The first 6 triangular numbers are shown.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T1** | **T2** | **T3** | **T4** | **T5** | **T6** |
|  |  |  |  |  |  |
| **1** | **3** | **6** | **10** | **15** | **21** |

You will be writing a program to produce the sequence of numbers on the bottom row of the table and the terms that follow in the sequence.

## Stage 1 – Study The Sequence

The triangular numbers are a sequence of numbers. It will be easier to write this program if you understand clearly how the sequence is formed.

1. What do you have to add to the first term to get the second term?
2. What do you have to add to the second term to get the third term?
3. Explain in words how the sequence is formed.

## Stage 2 – Counting From 1 To n

In this stage, the user is prompted to enter a number. Your program counts from 1 to that number. Here is the code that you need for the first stage. It will output the T numbers (top row of the table above) from 1 to the number you enter at the console.

 Dim n As Integer

 Console.Write("How many triangular numbers? ")

 n = Console.ReadLine()

 For i As Integer = 1 To n

 Console.WriteLine(i)

 Next

 Console.ReadLine()

1. Enter this code and show how you tested that the program worked.

## Stage 3 – Getting Closer

Now we want to change the output a little. If the user enters 3, we want the program to produce the following output,

T(1) =

T(2) =

T(3) =

At the end of each line, we will eventually want to display the triangular number. First though, just get the output in this format.

HINT: **“T({0}) =”, i**

## Stage 4 – Triangular Numbers

Declare a variable at the start of the program. It should be an integer called **triangle**.

**Before** the FOR ..NEXT loop**,** make this variable equal to 0.

**Inside** the FOR..NEXT loop, **before** the output, add the variable **i** to it. (triangle = triangle +i)

Output the triangular number at the end of the line so that now you get something like,

T(1) = 1

T(2) = 3

T(3) = 6

1. Make the change to the program to do this.
2. Show how you tested that the program produced the correct numbers.

## Stage 5 – nth Triangular Number

Now to change the program so that it only outputs the number that the user requests. If they enter 5, they should be told that the 5th triangular number is 15. There should be no other output.

1. Make sure that there are no output statements inside the loop. You still have the line that adds onto the variable triangle. **After** the loop has finished (after the word Next), output the variables **n** and **triangle**.
2. Show how you tested that the program worked at this stage.

## Stage 6 – Explore & Extend

The sequence of triangular numbers is very interesting to mathematicians.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Term Number** | **T1** | **T2** | **T3** | **T4** | **T5** | **T6** |
| **Triangular Number** | 1 | 3 | 6 | 10 | 15 | 21 |
| **Add To Previous** | 1 | 4 | 9 | 16 | 25 | 36 |
| **Cube Of Term Number** | 1 | 8 | 27 | 64 | 125 | 216 |
| **Square of Triangular** | 1 | 9 | 36 | 100 | 225 | 441 |

If add a triangular number to the previous triangular number, you get a square number. Add the 3rd triangular number to the second and you get three squared.

1. Copy your code and paste it into a new solution. Change the program so that it tells the user the triangular number that comes **before** the one that they ask for. You can do this easily by making the loop count from 1 to n-1 (n minus one). After the loop, make another variable (triangle2), equal to this number plus n. Now you have 2 triangular numbers. Add them together and output the result.
2. Explain what your program does now.

Make another program using your code from Stage 5.

1. Instead of outputting the triangular number that is requested, output the square of that number.
2. Explain what your program does now and the meaning of the number that you are displaying.
3. Compare your program and your approach with the program you wrote for or the example contained within the **Sum Of Cubes** task.