**Edinburgh College**

**NPA Software Development and Web**

**Software Design and Development**

**J27C76/008**

**Pyramid Height Calculator**

**Student Name: Adrián Sánchez Rodríguez**

**EC Number: EC1939656**

**Table of Content**

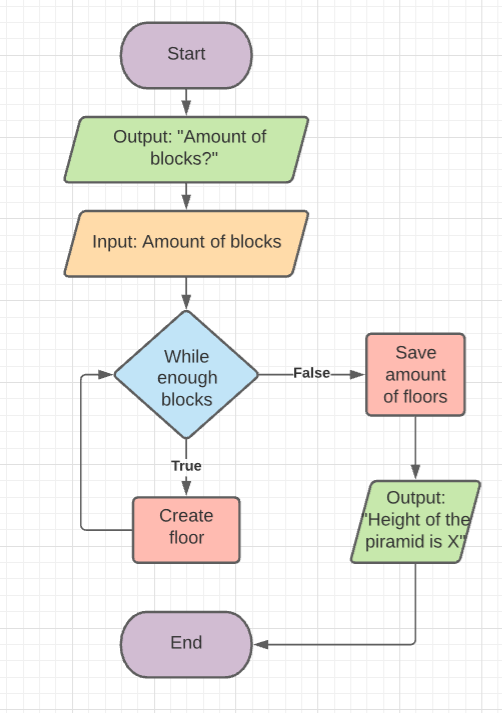
**1. Flowchart …………………………………………….……… pag.1**

**2. Pseudocode ………………………………………………… pag. 2**

**3. Source Code……………………………………………...… pag. 3**

**4. Test Scenario………………………………………….…… pag. 4**

**1- Flowchart**



**2- Pseudocode**

Display “Enter the number of blocks you would like your pyramid to have”;

Save the input and convert it to an integer;

While the number of blocks is higher than the number of blocks needed for the floor

{

Add a floor

Subtract the blocks used

Increase the number of blocks needed by 1

}

Display “The height of the pyramid is” and the number of floors.

**3- Source Code**

/\*

Program: Pyramid Height Calculator

Description: A program to calculate the number of floors a pyramid can have based on an user input.

Author: Adrian Sanchez (EC1939656)

Company: Edinburgh College

Version: 1

\*/

using System;

namespace Assesment9

{

class Program

{

static void Main(string[] args)

{

int floor = 0;

int counter = 1;

Console.WriteLine("Enter the amount of blocks you would like your pyramid to have:");

int input = Convert.ToInt32(Console.ReadLine());

while (counter<= input)

{

floor += 1;

input -= counter;

counter += 1;

}

Console.WriteLine("The height of the pyramid is:" + floor);

}

}

}

**4- Test Scenario**

|  |  |
| --- | --- |
| **Test case 1 – Adrian – 09/19/2020** | |
| Test Data | Input “6” |
| Expected behaviour | Output: “The height of the pyramid is 3” |
| Actual behaviour | Output: “The height of the pyramid is 3” |
| Comments / fixes | Worked as intended |

|  |  |
| --- | --- |
| **Test case 2 – Adrian – 09/19/2020** | |
| Test Data | Input “20” |
| Expected behaviour | Output: “The height of the pyramid is 5” |
| Actual behaviour | Output: “The height of the pyramid is 5” |
| Comments / fixes | Working as intended |

|  |  |
| --- | --- |
| **Test case 3 – Adrian – 09/19/2020** | |
| Test Data | Input “1000” |
| Expected behaviour | Output: “The height of the pyramid is 44” |
| Actual behaviour | Output: “The height of the pyramid is 44” |
| Comments / fixes | Working as intended |

|  |  |
| --- | --- |
| **Test case 4 – Adrian – 09/19/2020** | |
| Test Data | Input: “2” |
| Expected behaviour | Output: “The height of the pyramid is 1” |
| Actual behaviour | Output: “The height of the pyramid is 1” |
| Comments / fixes | Working as intended |

|  |  |
| --- | --- |
| **Test case 5 – Adrian – 09/19/2020** | |
| Test Data | Input: “shdjskdajkdh” |
| Expected behaviour | No output. Exception for a string stored in a int variable |
| Actual behaviour | Exception : System.FormatException: 'Input string was not in a correct format.' |
| Comments / fixes | This is because the program has no way to validate whether the input is a number or not, and tries to convert it to an integer, resulting in a crash. Further refactoring on how to identify the input as a string and add an option into the loop to let the user know that their input is invalid. |

|  |  |
| --- | --- |
| **Test case 6 – Adrian – 09/19/2020** | |
| Test Data | Input: “-1” |
| Expected behaviour | Output “The height of the pyramid is 0” |
| Actual behaviour | Output “The height of the pyramid is 0” |
| Comments / fixes | While this is not technically an error, an exception should be handled to specify that you can’t use negative numbers, since the height will always be 0 (never will have enough blocks to build at least 1 floor) |

|  |  |
| --- | --- |
| **Test case 7 – Adrian – 09/19/2020** | |
| Test Data | Input: “12.5” |
| Expected behaviour | No output. Exception for a double/float stored in a int variable |
| Actual behaviour | Exception : System.FormatException: 'Input string was not in a correct format.' |
| Comments / fixes | The application is trying to convert a floating number to an integer. Exception needed to handle non-integer numbers |

|  |  |
| --- | --- |
| **Test case 8 – Adrian – 09/19/2020** | |
| Test Data | Input: “999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999” |
| Expected behaviour | No output. The number will be too large for an 32b integer |
| Actual behaviour | Exception : System.OverflowException: 'Value was either too large or too small for an Int32.' |
| Comments / fixes | The number is too large to be stored in an Int32 variable. Exception to handle numbers that are longer that what could be stored, or store the variable in Int64 |