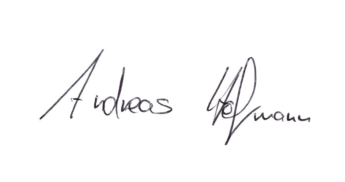


Homework 1

Programming C - IAX0583  
  
Andreas Hofmann - 177873

# Declaration of authorship

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. All works and major viewpoints of the other authors, data from other sources of literature and elsewhere used for writing this paper have been referenced.



Andreas Hofmann, 14.11.2017, Tallinn

Contents

[Declaration of authorship 2](#_Toc498432940)

[Variant: Method and function 4](#_Toc498432941)

[Brief overview of the task 5](#_Toc498432942)

[ArgoUML – Activity Diagram 6](#_Toc498432943)

[Program listings (code) with corresponding comments 7](#_Toc498432944)

[Complement the report with additional verbal description of your solution 10](#_Toc498432945)

[Snapshots 11](#_Toc498432946)

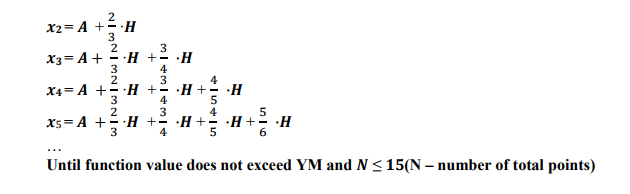
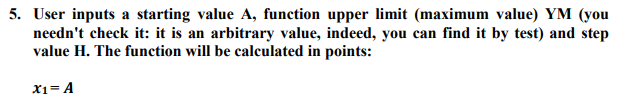
[Verification of results by means of MS Excel 13](#_Toc498432947)

# Variant: Method and function

In this task the method 5 and the function 32 was assigned to me which is:

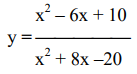


**Method 5:**



As you can see from the description, the user inputs a starting value A, also a function upper limit, which is the maximum value the function can obtain. In addition to that, the user inputs the step value H which is needed in order to calculate the value of x. Another restriction is the value N which is the total number of points and it is equal to 15, this number is not inputted by the user, it is given by the task.

**Function 32:**

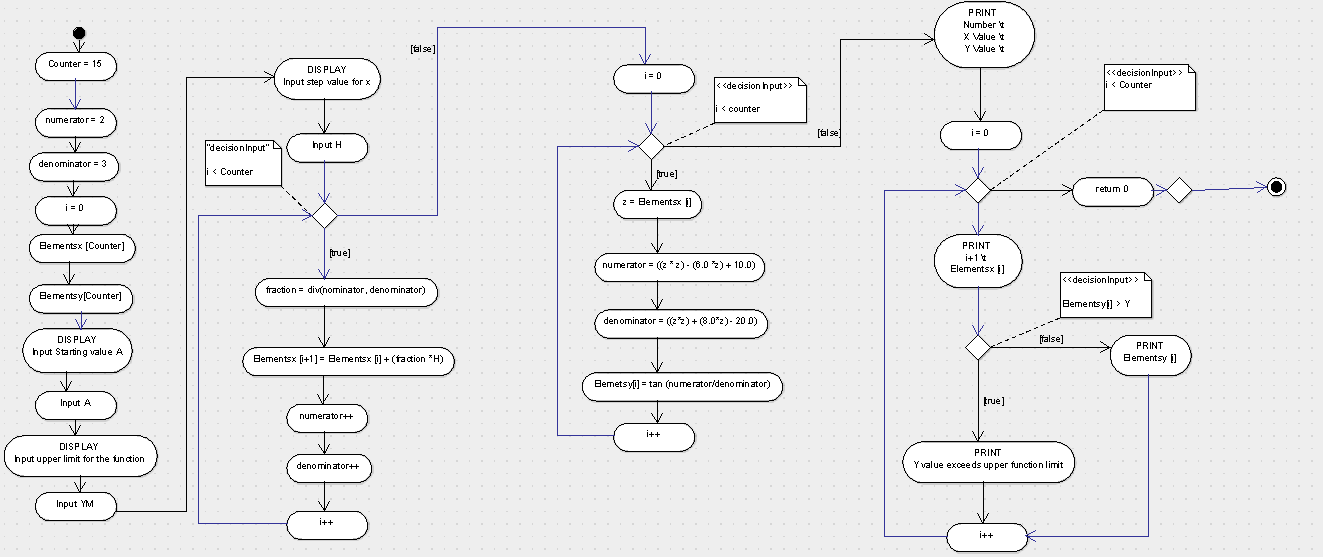


Here one can see, the function is an arbitrary y (x) function. In the nominator and in the denominator a binominal formula is used. In the nominator the second and in the denominator the first. However, one has to calculate first the value of x which is done with method 5 and then the corresponding y value.

# Brief overview of the task

In this task, one must first calculate the x value, which is done with the method given to us, and then the corresponding y value of the given function, which is the function 32.   
This task described above must be written as an algorithm in ArgoUML, JSP Editor, or yEd. In my case, I decided to create my activity diagram with ArgoUML. After the transformation of the task in an algorithm, one must transfer it to the corresponding programming code in language C. This also includes a heading paragraph in the code which gives information about the file name, authors name, update and a short description of the task. The program must display the user must input, and the output must be a table of the number in a consecutive flow (1, 2, 3, 4…), the x value and the corresponding y value.   
Furthermore, one has to create an excel file where the same task is done by hand in order to be safe that the program worked fine and has no errors, calculation, or language mistakes.

# ArgoUML – Activity Diagram



# Program listings (code) with corresponding comments

/\*

\* File: HW\_1\_Andreas\_Hofmann\_177873IASM\_5

\* Author: Andreas Simon Hofmann

\* Date: 17.10.2017

\* Update: 20.10.2017

\* Update: 25.10.2017

\* Update: 27.10.2017

\* Update: 01.11.2017

\* Update: 03.11.2017

\* Update: 07.11.2017

\* Description: calculates the function y = tan (x), with x as a continous value and y

\*/

#include <stdio.h>

#include <math.h>

#define Counter 15

int main (void)

//For this task we assume that the user inputs numbers

{

// declaration of the variables

float numerator, denominator, fraction, YM, A, z, H;

int i; // i is abbreviation for 'index', C for 'counter'

numerator = 2;

denominator = 3;

i = 0;

float Elementsx[Counter];

float Elementsy[Counter];

// Input by the user of the starting value

printf ("Input a starting Value: ");

scanf ("%f", &A);

printf ("The starting value is: %f\n\n",A);

Elementsx[i] = A;

// Input of the upper limit of the function

printf ("Input a function upper limit: ");

scanf ("%f", &YM);

printf ("The functions upper limit is: %f\n\n", YM);

// Input by the user of the step value of the calculation of x

printf ("Input a step value: ");

scanf("%f", &H);

printf("The functions step value is: %f\n\n", H);

// calculaing the x value of the function

// using auxiliary variables fraction, numerator, denominator. N is used as checking variable. i is used as index variable.

while (i < Counter)

{

fraction = numerator/denominator;

Elementsx[i+1] = Elementsx[i] + (fraction\*H);

numerator++;

denominator++;

i++;

}

// calculating the y value of the function with y = tan(x)

// i is used as index. z, numerator, denominator

for (int i = 0; i < Counter; i++)

{

z = Elementsx[i];

numerator = (z \* z - (6.0 \* z) + 10.0);

denominator = (z \* z + (8.0 \* z) - 20.0);

Elementsy[i] = numerator/denominator;

}

// Printing of the table with the Number (continously), the x value and the corresponding function value y

printf ("Number\t X Value\t Y Value\n\n");

// i is used as index. YM is the upper limit the function can obtain.

for (int i = 0; i < Counter; i++)

{

printf("%d\t %f\t",i+1, Elementsx[i]);

if (Elementsy[i]>YM)

printf("exceeds upper Y limit\n");

else

printf ("%f\n", Elementsy[i]);

}

return 0;

}

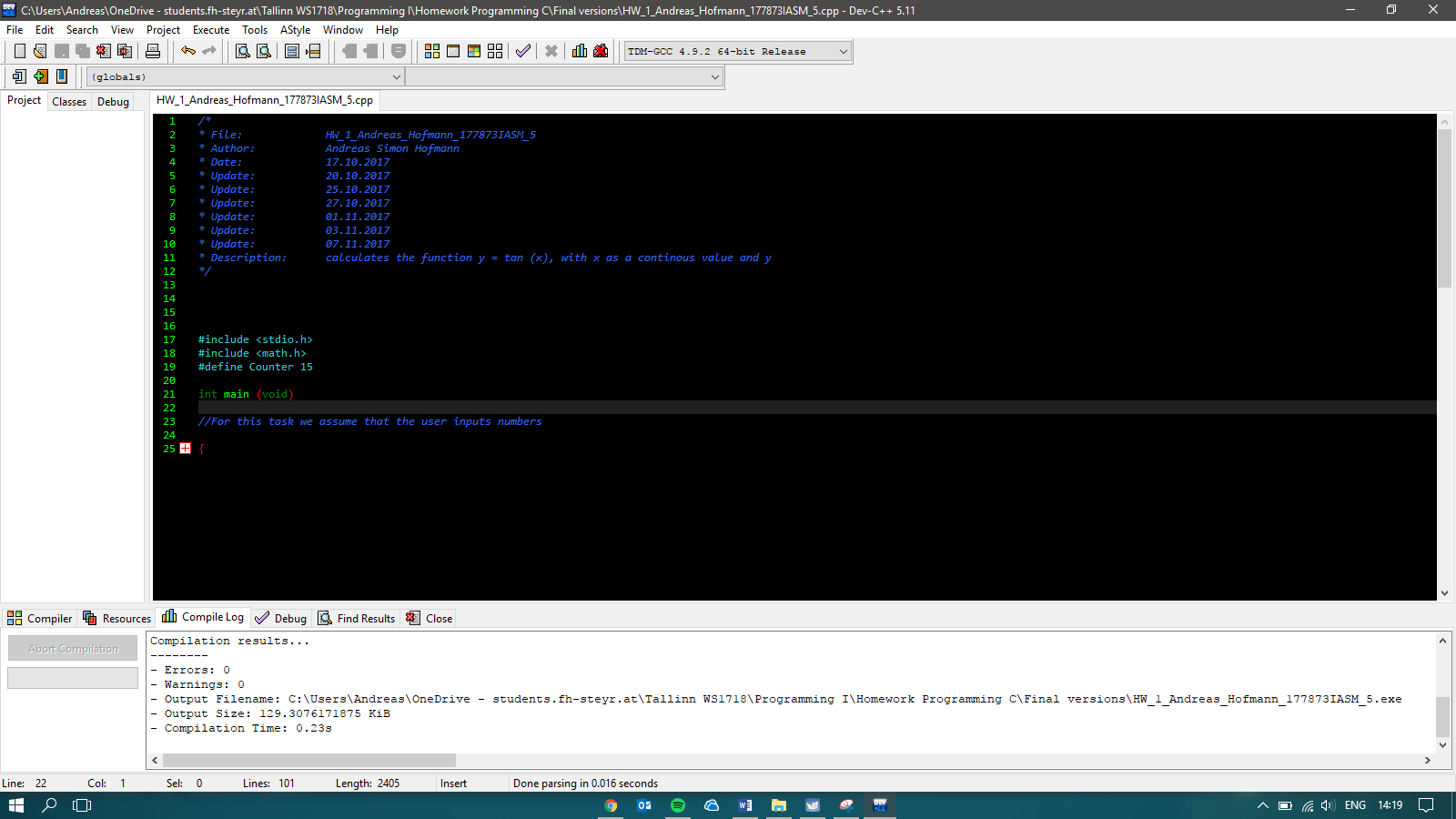
# Complement the report with additional verbal description of your solution

In this task, I used two arrays (‘Elementsx’ and ‘Elements’), because, in my opinion, it is easier to display and calculate with them. However, at the beginning, the ‘Elementsx’ array caused trouble because I had to somehow convert it to a single integer in order to calculate the ‘Elements’ array since the program displayed an error if I tried to calculate with it. Therefore, I created a for the statement, which assigns the auxiliary variable float z (line 82) to the ‘Elementsx’ with the index [i]. Since this gives empowers me to calculate with the single ‘Elementsx’, it was then easy to get the corresponding y value.  
In line 46 I assigned the input value of the user ‘A’ to the array ‘Elementsx’ since it has to be the starting value and all other values depend on this value.

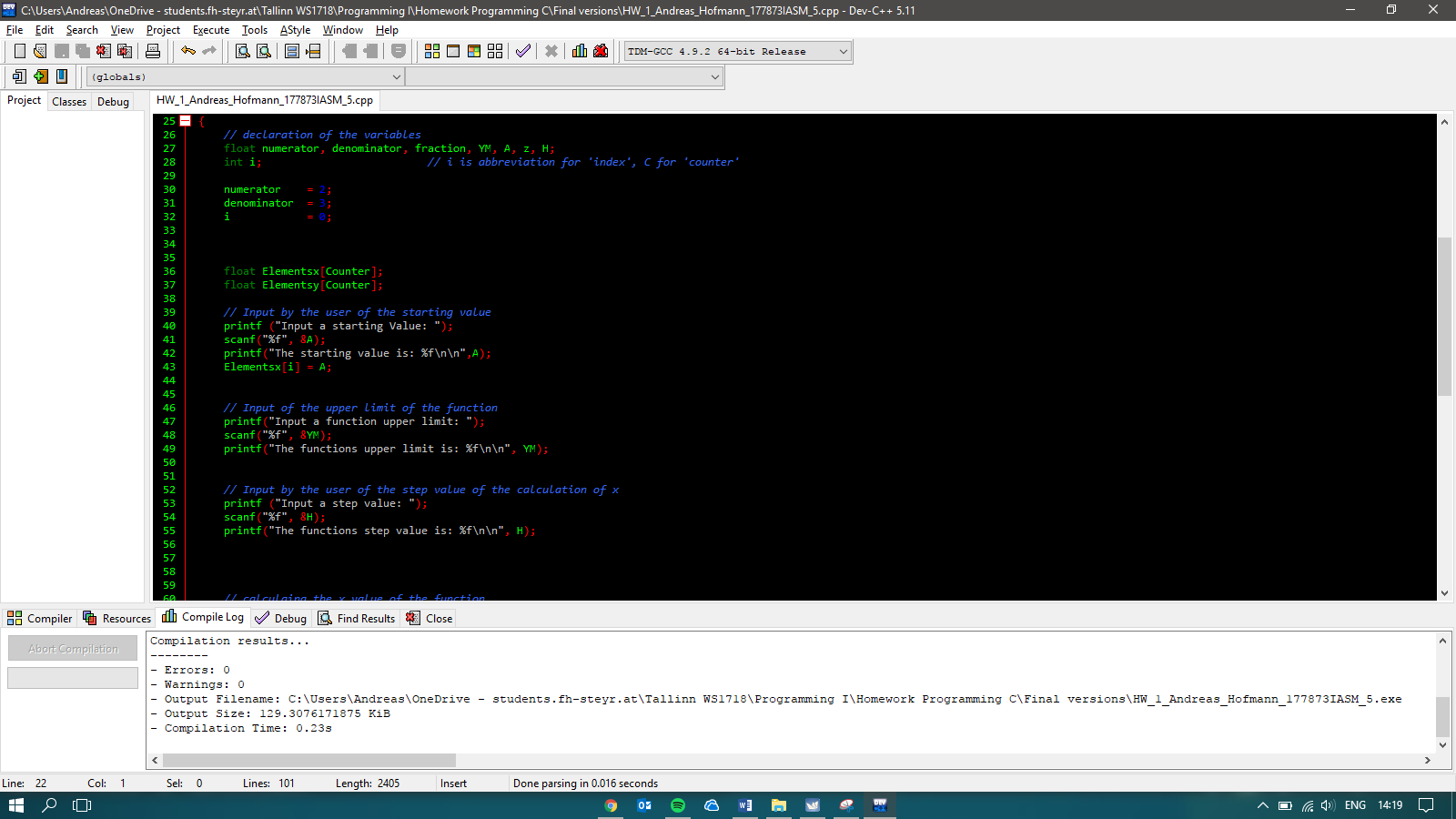
In the while loop, line 71 it states Elements [i+1]. This i+1 is necessary because if this is not the case, not the next number in the array will be calculated and this brings a problem in the array so that the numbers and values would have a problem.

In line 99 I implemented an if statement in order to find out if a value of the array ‘Elements’ exceeds the upper limit of the function which was input by the user. If it is the case that this value exceeds the input value, the table will display ‘exceeds upper Y limit’. If it is not the case, the number will be displayed correctly.

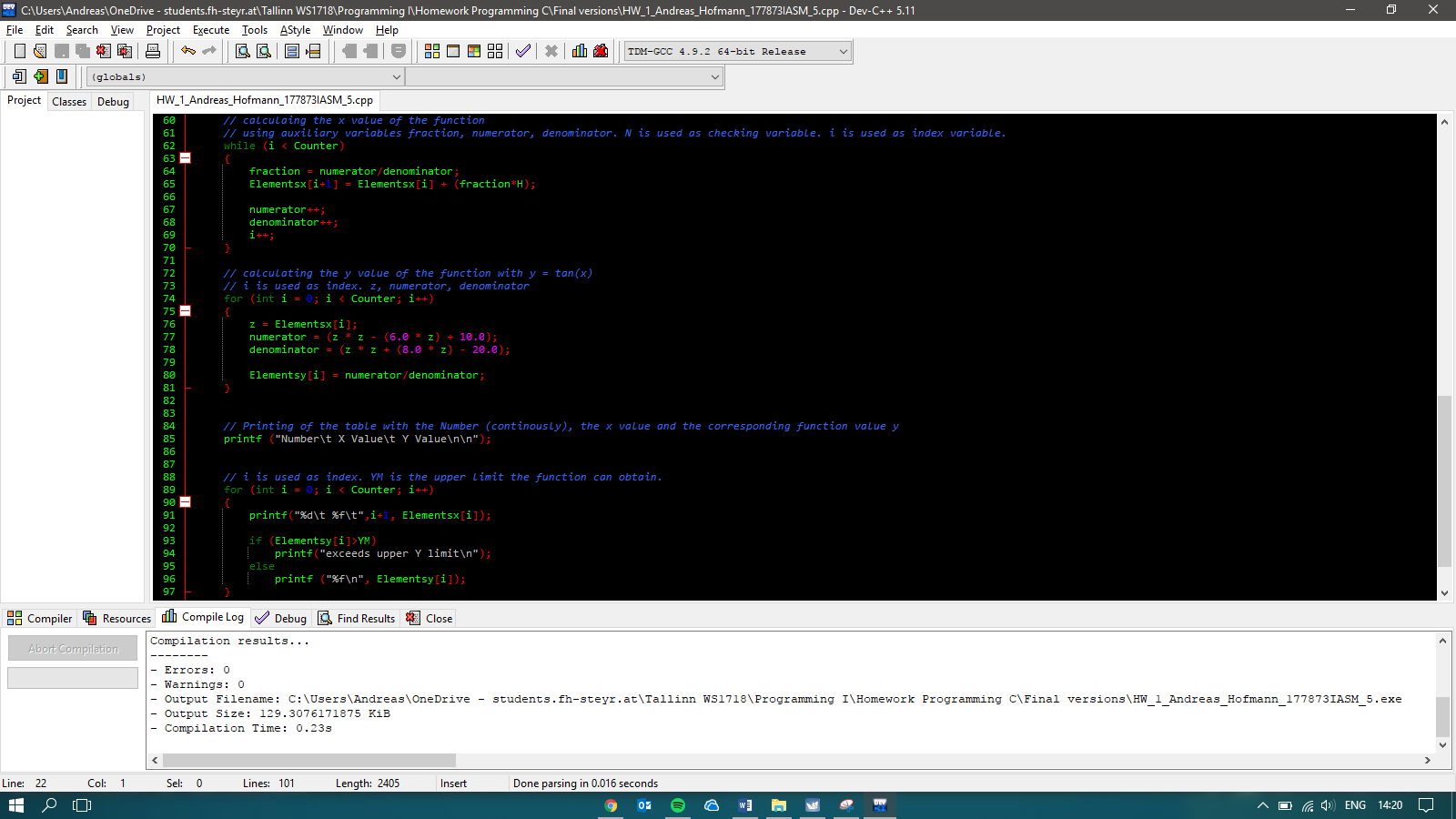
# Snapshots



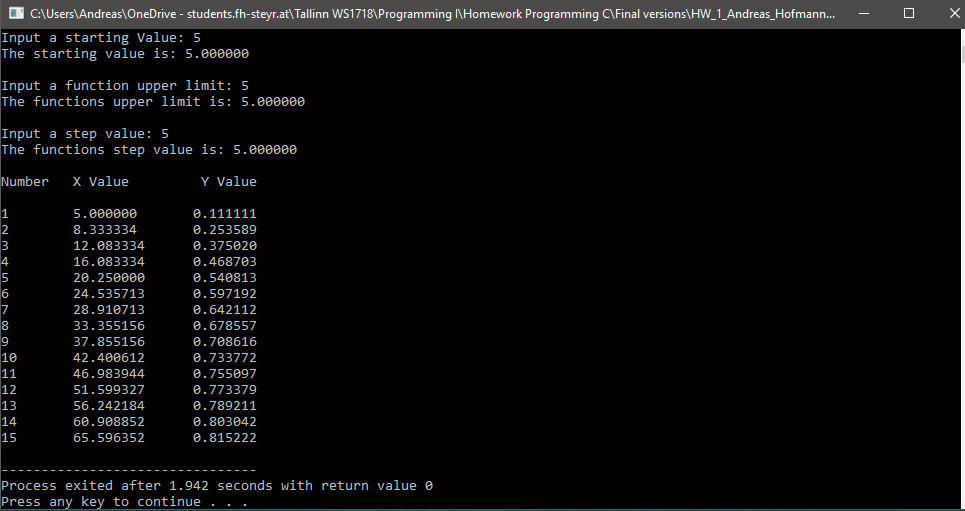
The header of the program which gives a brief description of the file, updates, author, and description of the file.



This snapshot shows the program code that I have used in order to define the input of the user.



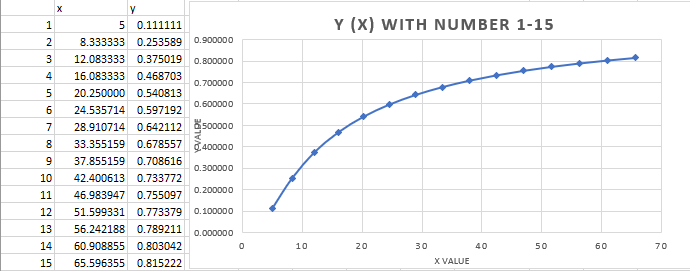
Snapshot of the rest of the program. The while loop is for calculating the x values with method 5. The first for loop is for calculating the y value, and the second for loop is for the desired output, which is displaying the number, the x value and the corresponding y value in a table. Furthermore, I used an if statement in order to check if the value does not exceed the upper y limit (YM), which was input by the user.

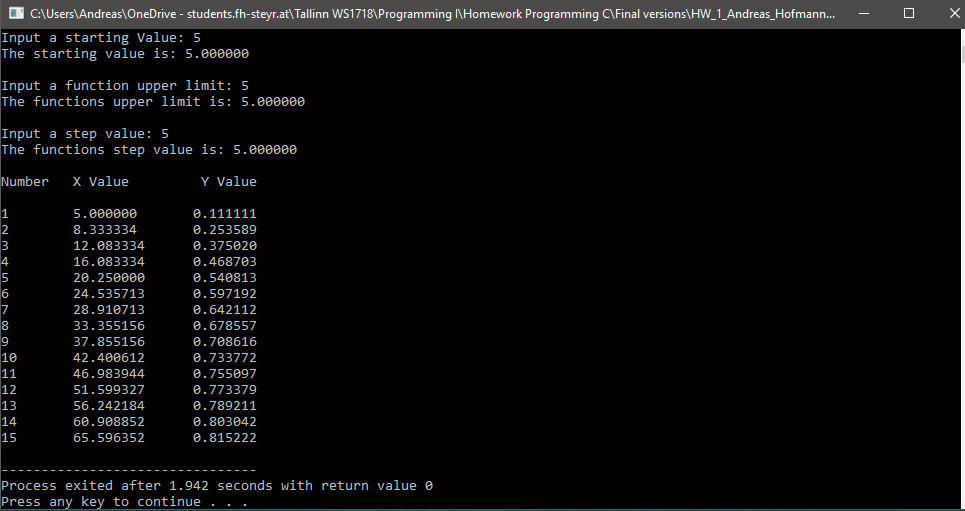


Snapshot of the table by using a starting value, upper y limit, and step value of **5**.

# Verification of results by means of MS Excel

In the following snapshot of excel and the output of the program, one can see that the numbers of MS Excel and the program are the same.





For further analysis, the excel chart from 16 – 50 displays the same as the output of the program, when changing the end value to 50

