

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems
Study Centre for Computer Systems

Sigita Ševceva 204434IV
IAX0583 Programming I

Tabulating a function

$$y = 1 - \frac{1 - \sqrt{4 - x^2}}{40x^2 + \sqrt{x}}$$

Homework I

Supervisor Risto Heinsar

Engineer

Tallinn 2020

Declaration of originality

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.

Name: S.Ševceva

Date: 7.10.2020

Table of contents

Declaration of originality	2
Table of contents	3
Task description	4
Function plot.....	5
Program description	6
Algorithm.....	7
Screenshots	8
References	9

Task description

Method

6. User inputs a starting value A, step H and upper limit of the function value YM.

The following conditions have to be true: $H > 0$.

The function value y will be calculated in the following points:

A

A + H

A + 2H

A + 3H

...

while the condition $y < YM$ holds true, however not more than 15 times.

Function

$$31. \quad y = 1 - \frac{1 - \sqrt{4 - x^2}}{40x^2 + \sqrt{x}}$$

Function plot

To provide a better insight into my homework, I have included two graphs with different scales below, by using *Grapher* (Apple, 2005-2019) software for visual plot functioning.

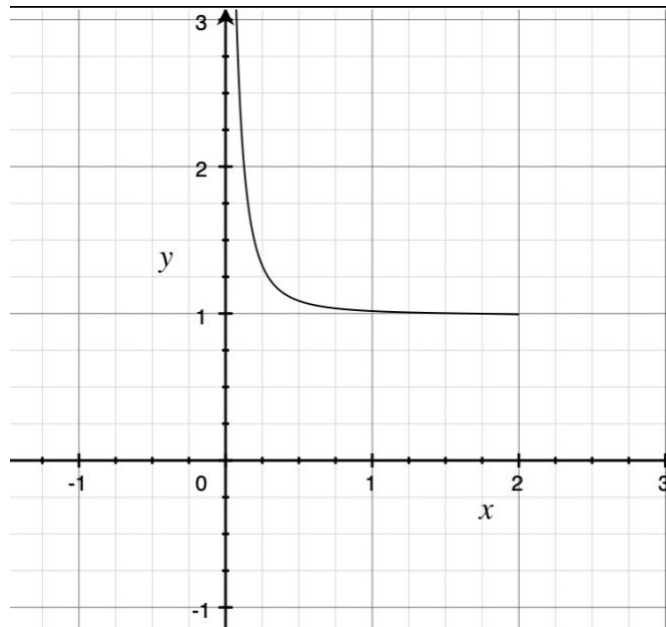


Figure 1: Function $y = 1 - \frac{1 - \sqrt{4 - x^2}}{40x^2 + \sqrt{x}}$ graph around zero

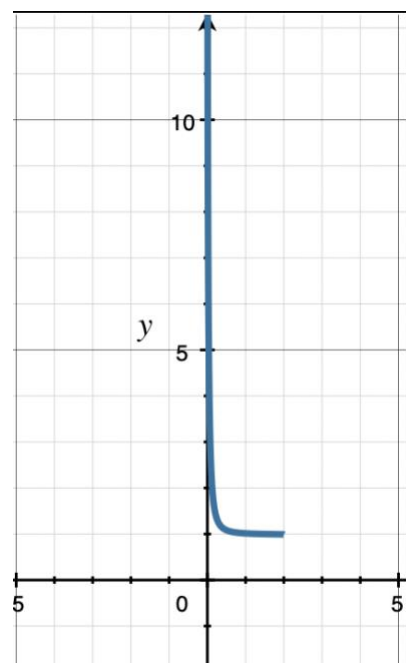


Figure 2: $y = 1 - \frac{1 - \sqrt{4 - x^2}}{40x^2 + \sqrt{x}}$ graph

Program description

The C language program is made in *Geany* IDE using standard C libraries `<stdio.h>` and `<math.h>` for mathematical operations. Its goal is to calculate values of a given mathematical equation by taking into account information by user input.

Regarding the program flow, firstly, 3 macros are defined, namely, the spacing of final table columns. Then, the main function is created and 11 variables are defined: an array of x floating numbers (holding up to 15 values); similarly, an array of y values; an integer number of user input for number of calculations to be done (`times`); the function starting value stored in variable `A`; function argument step stored in `H`; the upper limit of y value stored in (`Ymax`); two counters `i` and `n` for x and y values; enumeration variable that will be used in the final table columns (`num`), as well as two character strings to inform the user about missing values or values exceeding `Ymax`. All variables that will be directly used in calculations are set as floating-point numbers for precision; the rest are assigned as integers or chars where necessary.

Secondly, the user is introduced to the program functions using `puts` functions. Then, as the user shall not exceed the number 15 when providing the number of calculations, a *do while* loop is introduced to handle cases when the user inputs a number greater than 15. Next, a *do while* loop is used to prevent the user for inputting anything other than numerical value for x_0 value, as well as `Ymax`. This is done by validating the input with `getchar() != '\n'`. This also prevents the user from complex number input. Similarly, the given task requires for the step to be over 0, thus a *do while* loop is used to not allow the user to input `H` smaller or equal than 0.

Next, the program executes calculations to find each x value starting from x_0 defined as `A`, adding a step `H` multiplied by the counter `i`. The counter is incremented after each executed loop until it reaches the number of times chosen by the user. Similarly, a *for loop* is introduced to calculate each y value according to the already calculated and stored x values. Here *math.h* standard library comes into use by calculating powers and square roots of the given mathematical equation. Following is a table header which is created with three columns: for numeration, x and y values. The spacing is automatically assigned by the previously defined macros of length.

Finally, the x and y values are displayed in the table, reinitializing the previously introduced counters to 0 and incrementing them until user specified (`times`). Values are aligned by using defined *macros* of length and .3 to display 3 numbers after the comma. If the y values exceed user inputted `Ymax`, the function prints "`Ymax`", as well as "`No val`" in case of missing or non-computational values. The program returns 0 upon successful termination.

There is a limitation to this given mathematical equation, namely, as seen in the *Figures 1 and 2*, x values can only be given in the range from 0-2, as the corresponding y values do not exceed these x values. For this reason, a small amplitude of values and steps should be inputted for the y calculations to return real values, because the closer to 0 get the x values, the closer to infinity get y values.

Algorithm

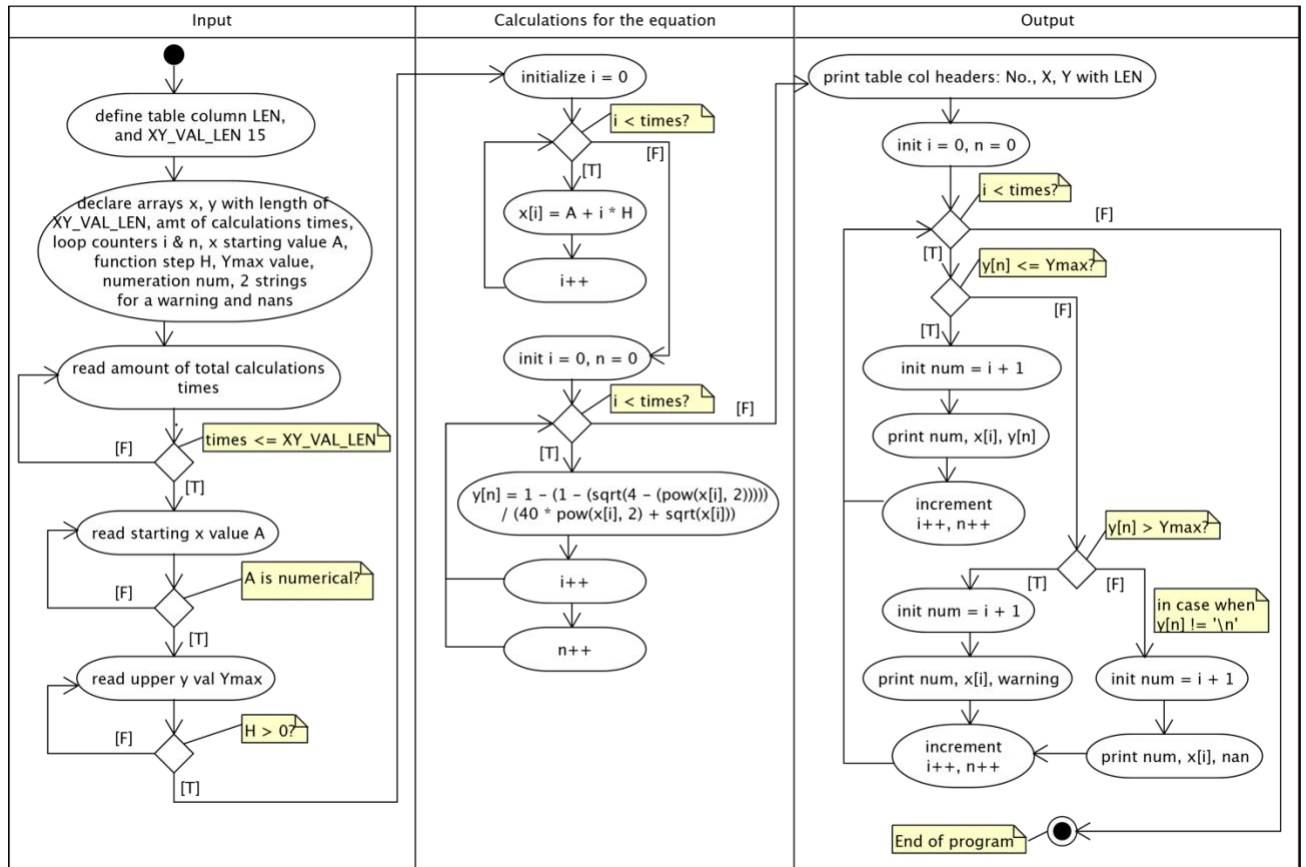


Image 1: Algorithm of program

Screenshots

```

sigitasevceva — geany_run_script_41FUR0.sh — geany_run_script_41FUR0.sh...
This program will calculate value y for the function
y = 1 - (1 - sqrt(4 - x^2)) / (40x^2 + sqrt(x))
up to 15 pts. The real life function would be [0; inf]
How many pts should function calculate? (0-15)
15
Please enter function starting value (x0):
0
Please enter upper limit of function value (Ymax):
1000
Please enter function argument step: (H>0)
0.1
No.    X      Y
1      0.000  inf
2      0.100  2.393
3      0.200  1.484
4      0.300  1.236
5      0.400  1.136
6      0.500  1.087
7      0.600  1.060
8      0.700  1.043
9      0.800  1.031
10     0.900  1.024
11     1.000  1.018
12     1.100  1.014
13     1.200  1.010
14     1.300  1.008
15     1.400  1.005

(program exited with code: 0)
Press return to continue

```

Figure 2: Program functioning properly

```

This program will calculate value y for the function
y = 1 - (1 - sqrt(4 - x^2)) / (40x^2 + sqrt(x))
up to 15 pts. The real life function would be [0; inf]
How many pts should function calculate? (0-15)
15
Please enter function starting value (x0):
1
Please enter upper limit of function value (Ymax):
1
Please enter function argument step: (H>0)
0.1
No.    X      Y
1      1.000  >Ymax
2      1.100  >Ymax
3      1.200  >Ymax
4      1.300  >Ymax
5      1.400  >Ymax
6      1.500  >Ymax
7      1.600  >Ymax
8      1.700  >Ymax
9      1.800  0.999
10     1.900  0.997
11     2.000  0.994
12     2.100  >Ymax
13     2.200  >Ymax
14     2.300  >Ymax
15     2.400  >Ymax

```

Figure 1: Program found some invalid values (larger than user input Y max)

```

Please enter function starting value (x0):
3
Please enter upper limit of function value (Ymax):
100
Please enter function argument step: (H>0)
2
No.    X      Y
1      3.000  No val
2      5.000  No val
3      7.000  No val
4      9.000  No val
5      11.000 No val
6      13.000 No val
7      15.000 No val
8      17.000 No val
9      19.000 No val
10     21.000 No val
11     23.000 No val
12     25.000 No val
13     27.000 No val
14     29.000 No val
15     31.000 No val

```

Figure 5: Program found no values for y because the given function does not return values for $x > 2$ (see Figures 1 & 2)

References

- Apple Inc. (2005-2019) *Grapher*, software, Version 2.7.
The Geany contributors (2005) *Geany*, software, Version 1.36.