TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Emma Hovorkova 243116IV

PROGRAMMING I HOMEWORK 1

Tallinn 2024

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Emma Hovorkova

02.10.2024

Table of Contents

List of Figures

List of Tables

1. Homework definition

According to your student code choose a function argument (x) finding method and according to your student code choose the function $y = f(x)$. Come up with an algorithm to solve the task and implement code in C that corresponds to the algorithm. The homework should include both, the algorithm and the code. All of the input data should be inserted from keyboard and it can any real number. Results will be outputted on the display as a table with arguments in the first column and corresponding function values in the second column. Function value should be displayed only if it exists i.e. it is final and real number. Else if function value is not defined (it is infinite) or it is complex number, program should display 'not available' or 'complex number'. For complex number you can also display the value in the form of $a + bi$, where a is the real part and b is the imaginary part.

1.1 Method and function

Since my student code is 243116IV I was given a method 4 and function 19.

1.1.1 Method

User inputs a starting value A, stopping value B, a step H, and the step coefficient C. The following conditions have to be true: $A < B; H > 0; C \geq 1$. The function value y will be calculated in the following points:
$x_1 = A$
$x_2 = A + H$
$x_3 = A + H + CH$
$x_4 = A + H + CH + C^2H$
...
while $x_N < B$ and $N < 15$.

1.1.2 Function

$y = \frac{6x^2 - \sqrt{\frac{1+x^2}{4-x^2}}}{8-x^3}$

2. Function analysis

Here I present the mathematical properties of given method and function.

2.0.1 Method

User inputs a starting value A, stopping value B, a step H, and the step coefficient C. The following conditions have to be true: $A < B; H > 0; C \geq 1$. The x_n for $n = 0$ is $x_0 = A$, for $x = 1$ is $x_1 = A + H$ and for $n > 2$ it can be counted using following formula
$x_n = x_{n-1} + C^{x-1}H$
The outcome is counted until $x_n < B$ or $n < 15$.

2.0.2 Function

The function is $y = \frac{6x^2 - \sqrt{\frac{1+x^2}{4-x^2}}}{8-x^3}$. We can see the graph in 1. Since the value gets high for $1.9 < x < 2$, there is the larger graph in 2.

The domain for this function the domain is $-2 < x < 2$ and the range is $y \leq 135.232$. The highes value of y is when $x = 1.9951$ (counted using WolframAlpha, see `https://www.wolframalpha.com/`).

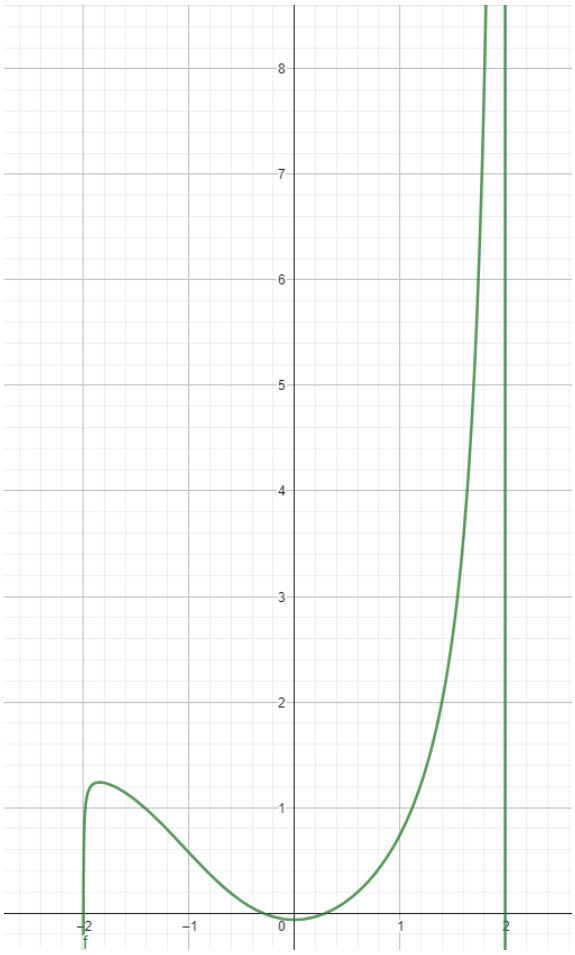

Figure 1. Function graph closer

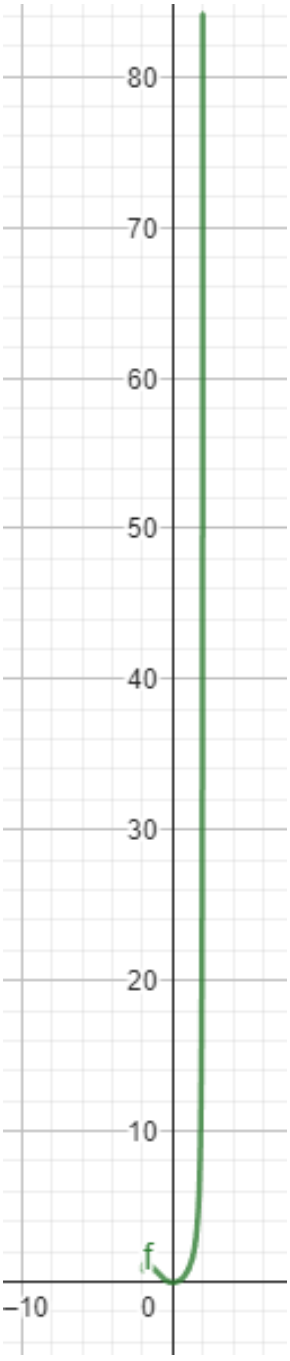

Figure 2. Function graph larger

3. Solution description

In this section I describe how my solution works. The algorithm is divided into 4 phrases. First is loading input and checking it as well. Second is counting the values of the method. Then counting the values of function and final phase is printing the values into console.

For UML diagram see 3.

3.1 Loading input

The algorithm starts by loading four numerical inputs: A, B, H, and C. These inputs are subjected to initial validation checks. The algorithm first verifies that A is less than B. Then it checks whether H is non-negative and whether C is greater than or equal to 1. If any of these conditions is not true, the algorithm asks for the inputs again. Once these conditions are satisfied, the algorithm seeks user confirmation to proceed. If the user does not confirm, the algorithm asks for the inputs again; otherwise, it moves on to the next phase.

3.2 Counting method

This part begins by initializing the variable n to 0 and setting x to the value of A. A loop is then entered, which continues as long as x is less than B and the number of iterations (n) is less than 15. During each iteration, n is incremented, and the current value of x is stored in the array args[n]. Depending on the value of n, different updates to x are made as explained in previous chapter: if n is 0, x is set to A; if n is 1, x is set to $A + H$; for all other values of n, x is calculated as args[$n - 1$] + $C^{n-1}H$. This process continues as long as the updated x remains less than B. At the end of the loop we check the condition $x < B$ for the last value of x and if it is not satisfied, n is decremented by 1 to ignore the last value.

3.3 Counting function values

In this phase a new variable i is initialized to 0, and the algorithm enters another loop that continues while $i < n$. During each iteration, it retrieves a value x from the args array. If the value of x falls outside the range [-2, 2] the corresponding function value stored in the array *fcn* is marked as undefined. However, if x lies within the acceptable range,

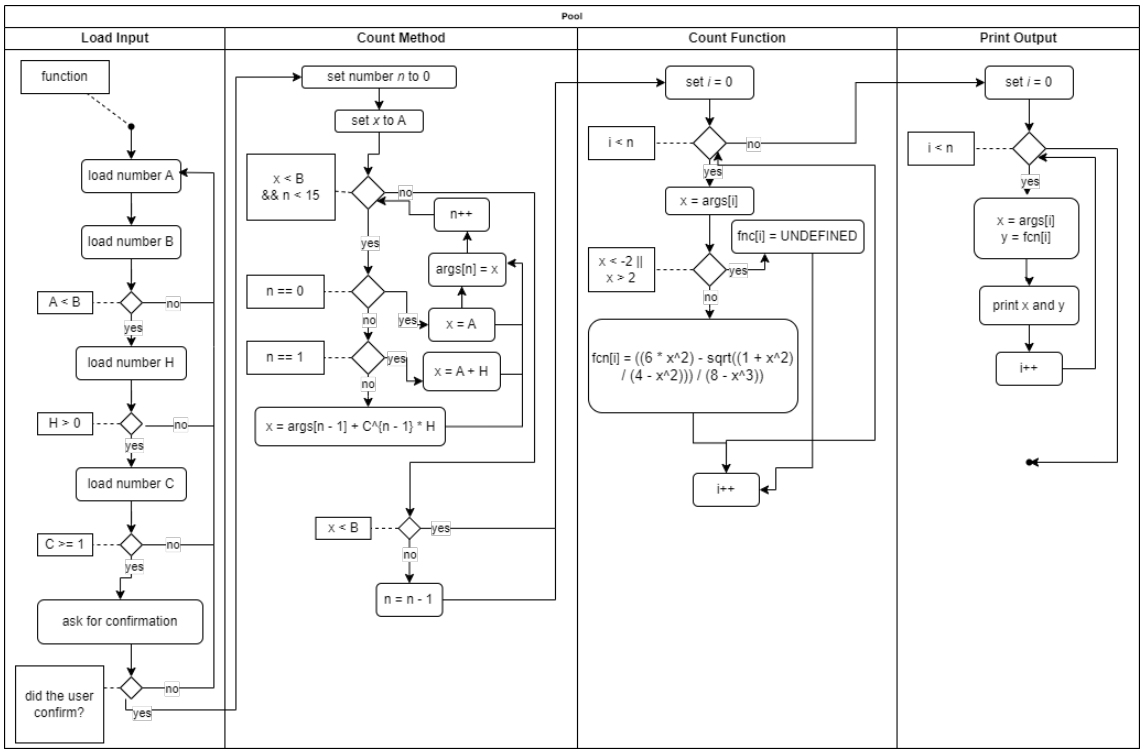

Figure 3. UML Diagram

the function value stored into fnc[i] is computed as fnc$[i] = \frac{6x^2 - \sqrt{\frac{1+x^2}{4-x^2}}}{8-x^3}$ The value of i is incremented after each iteration, and the loop continues until all elements in the args array have been processed.

3.4 Printing output

Finally, the algorithm moves to the "Print Output" phase, where the computed values are printed. A variable i is once again initialized to 0, and a loop runs until $i < n$. For each iteration, the corresponding values of x (from args[i]) and y (from fnc[i]) are printed. The value of i is incremented after each iteration until all results have been displayed, concluding the algorithm.

4. Implementation and testing

Here I mention few implementation details and describe each function. Then I show how I tested my program.

4.1 Implementation

At the beginning of the code the constant *UNDEFINED* is set to value 1000. It is later used for cases when the value of function is undefined (when the argument x is either $x < -2$ or $x > 2$). I chose the value 1000, since the function can never reach the value (it has a range $y \leq 135.232$).

The other constant defined at the beginning is *EPSILON* set to value 1e-9. This is the setting of precision for comparing double numbers. Because of how doubles are represented in computer, we can't compare them as integer numbers. To compare them I use my own functions:

```
int isEqual(double a, double b){
  return fabs(a - b) < EPSILON;
}

int isLess(double a, double b){
  return (b - a) > EPSILON;
};

int isLessThan(double a, double b){
  return (isLess(a, b) || isEqual(a, b));
}
```

The code is divided into several functions.

4.1.1 getParameter

```
double getParameter(char name);
```

This function takes one argument *name*.

This function give one output - the number it loaded from user.

Then it prints "Please give me *name*" and using the*scanf* function reads the input. It reads the input until it is a real number and then returns the number.

4.1.2 loadInput

```
void loadInput(double *A, double *B, double *H, double *C);
```

This function takes 4 arguments, all of them are pointers to double numbers *A, B, H, C*.

This function gives no output.

The function manages the process of collecting four key parameters: A, B, H, and C, which represent the starting value, stopping value, step, and coefficient, respectively. The function continuously prompts the user to input these values while ensuring that specific conditions are met — A must be less than B, H must be positive, and C must be greater than or equal to 1. After receiving valid input, the function asks the user to confirm whether to proceed with these values. If the user confirms (by entering 1), the function terminates; otherwise, the process starts over, allowing the user to input new values.

4.1.3 countMethod

```
int countMethod(double *args, double A, double B, double H, double C);
```

This function takes 5 parameters: array of doubles args and 4 numbers A, B, H and C.

This function give an output of how many values were added to the array args.

The function is where the algorithm generates a series of arguments based on the input parameters A, B, H, and C. Starting with an initial value of A, it creates subsequent arguments using a formula that incorporates the coefficient C and the step value H. The function continues generating values until either A reaches or exceeds B, or the number of generated arguments reaches 15. The generated values are stored in the array args, and the total number of arguments is returned.

4.1.4 countFunction

```
void countFunction(int n, const double *args, double *fnc);
```

This function takes a number n represting how many argumets are in the array args, the array args and the array fcn to store the values.

```
Please input the starting value A, stopping value B, step H and step's coefficient C
Please give me A
1.2
1.2
Please give me B
10.8
10.8
Please give me H
2.3
2.3
Please give me C
2
2
I have recieved starting value 1.200000, stopping value 10.800000, step 2.300000 and step's coddicient 2.000000
Do you want to count the method with these parameters? Write 1 for yes
1
1
argument | function
1.200000 | 1.221894
3.500000 | UNDEFINED
```

Figure 4. Output in console

This function gives no output.

The function takes the generated arguments and computes the values of a mathematical function for each argument. The function iterates through the args array and, for each argument, checks if it falls within a valid range (between -2 and 2). If the argument is outside this range, the corresponding result is set to UNDEFINED. If the argument is within range, the function applies a specific mathematical formula to calculate the result and stores it in the fnc array.

4.1.5 printResults

```
1 void printResults(int n, const double *args, const double *fnc);
```

This function takes a number n representing the lenght of both arrays and an array args and fnc.

This function gives no output.

Finally, the printResults function displays the arguments and their corresponding function results in a tabular format. It iterates over the args and fnc arrays and prints each argument alongside its result. If a result was marked as UNDEFINED during the calculation phase, the function prints "UNDEFINED" instead of a numerical value. This function provides a clear, formatted output of the results, making it easy for the user to review the computed values.

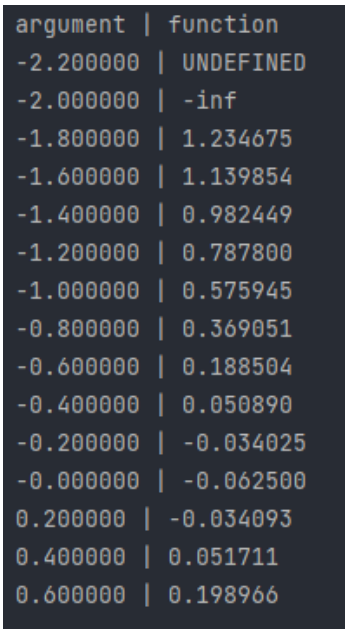

Figure 5. Ouput for Set 1

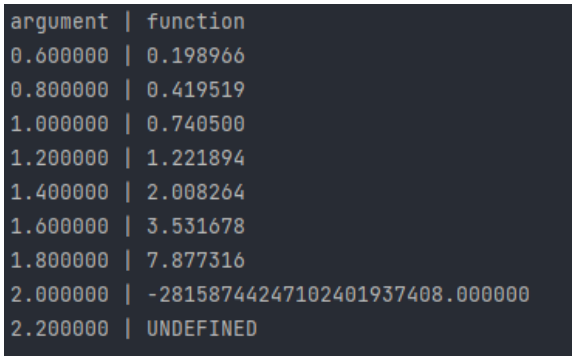

Figure 6. Output for Set 2

You can see the console output in 4.

4.2 Testing

Here I describe my testing sets of data.

4.2.1 Set 1

First testing set is to get values for many arguments. I set $A = -2.2$, $B = 2.2$, $H = 0.2$ and $C = 1$. This gives me values for arguments $x \in \{-2.2, -2, -1.8, ..., 0.6\}$. See output in 5.

-1,8	1,234675
-1,6	1,139854
-1,4	0,982449
-1,2	0,787800
-1	0,575945
-0,8	0,369051
-0,6	0,188504
-0,4	0,050890
-0,2	-0,034025
0	-0,062500
0,2	-0,034093
0,4	0,051711
0,6	0,198966
0,8	0,419519
1	0,740500
1,2	1,221894
1,4	2,008264
1,6	3,531678
1,8	7,877316

Figure 7. Results for the function in MS Excel

```
Please input the starting value A, stopping value B, step H and step's coefficient C
Please give me A
1
1
Please give me B
0
0
Value A cannot be bigger than B. Try again.
```

Figure 8. Output for Set 3.1

4.2.2 Set 2

Because I only got 15 arguments for previous set, this set is created to gather rest of the arguments $x \in \{0.6, ..., 2.2\}$. I set $A = 0.6$, $B = 2.2$, $H = 0.2$ and $C = 1$. See output in 6.

In order to check the values for $x \in \{-0.8, ..., 1.8\}$ I used MS Excel to count the values. See the results in 7.

4.2.3 Set 3

This is a set to test wrong inputs

```
Please give me A
0
0
Please give me B
2
2
Please give me H
-1
-1
Value H cannot be negative number. Try again.
```

Figure 9. Output for Set 3.2

```
Please input the starting value A, stopping value B, step H and step's coefficient C
Please give me A
0
0
Please give me B
2
2
Please give me H
1
1
Please give me C
0
0
Value C cannot be less than 1. Try again.
```

Figure 10. Output for Set 3.3

```
Please give me A
test
test
Uknown value. Please give me a real number
```

Figure 11. Output for Set 3.4

```
Do you want to count the method with these parameters? Write 1 for yes
0
0
Understand. Give me new values then
```

Figure 12. Output for set 3.5

```
Do you want to count the method with these parameters? Write 1 for yes
test
test
Understand. Give me new values then
```

Figure 13. Output for set 3.6

1. $A = 1$ and $B = 0$. See output in 8
2. $A = 0$, $B = 2$ and $H = -1$. See output in 9
3. $A = 0$, $B = 2$, $H = 1$ and $C = 0$. See output in 10
4. type *test* instead of number when asked for A. See output in 11
5. type 0 when asked for confirmation. See output in 12
6. type *test* when asked for confirmation. See output in 13

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Emma Hovorkova

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my homework "Programming I homework 1".
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

02.10.2024

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.