Numerical Methods

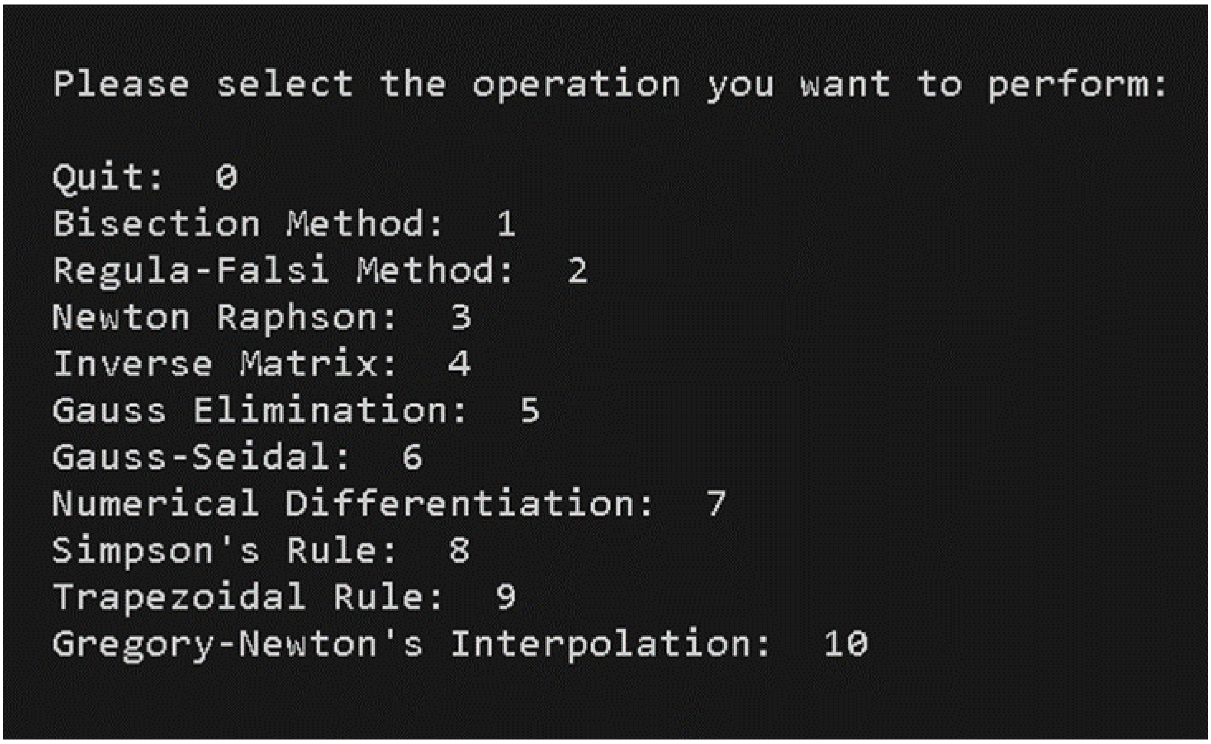
Preface

The program is designed to perform 10 specific operations. These operations are as follows:

1. *Bisection method*
2. *Regula-Falsi method*
3. *Newton-Rapshon method*
4. *Inverse of an NxN matrix*
5. *Gauss elimination method*
6. *Gauss-Seidel method*
7. *Numerical differentiation*
8. *Simpson method*
9. *Trapez method*
10. *Gregory-Newton interpolation*

Main Menu

When the program is executed, the user is presented with methods/applications to choose from. The user can run the program by entering the number next to the method/application of choice. The program repeats itself until the user enters 0.



```
Please select the operation you want to perform:

Quit:  0
Bisection Method:  1
Regula-Falsi Method:  2
Newton Raphson:  3
Inverse Matrix:  4
Gauss Elimination:  5
Gauss-Seidal:  6
Numerical Differentiation:  7
Simpson's Rule:  8
Trapezoidal Rule:  9
Gregory-Newton's Interpolation:  10
```

Supported Functions

Methods for finding roots (Bisection, Regula-Falsi, Newton-Raphson), numerical differentiation, and integral methods (Numerical Differentiation, Simpson’s Rule, Trapezoidal Rule). These functions can be adjusted to accommodate polynomial, exponential, logarithmic, trigonometric, and inverse trigonometric function types, respectively. For each function type, after entering the number of expressions of that type, the parameters of that type are entered as many times as specified. The parameters for these types are as follows:

Polinom

$$x_{coef} \times x^{x_{\mathrm{exp}}}$$

Exponential

$$fn_{coef} \times \left(base^{x_{coef} \times x^{x_{exp}}}\right)^{fn_{exp}}$$

Logaritmic

$$fn_{coef} \times \log_{base}\left(\left(x_{coef} \times x^{x_{ecğ}}\right)^{fn_{ecp}}\right)$$

Trigonometric

$$fn_{coef} \times trig_{fn}\left(x_{coef} \times x^{x_{\mathrm{exp}}}\right)^{fn_{\mathrm{exp}}}$$

$$trig_{fn} = \begin{cases} \sin & 0 \\ \cos & 1 \\ \tan & 2 \\ \cot & 3 \end{cases}$$

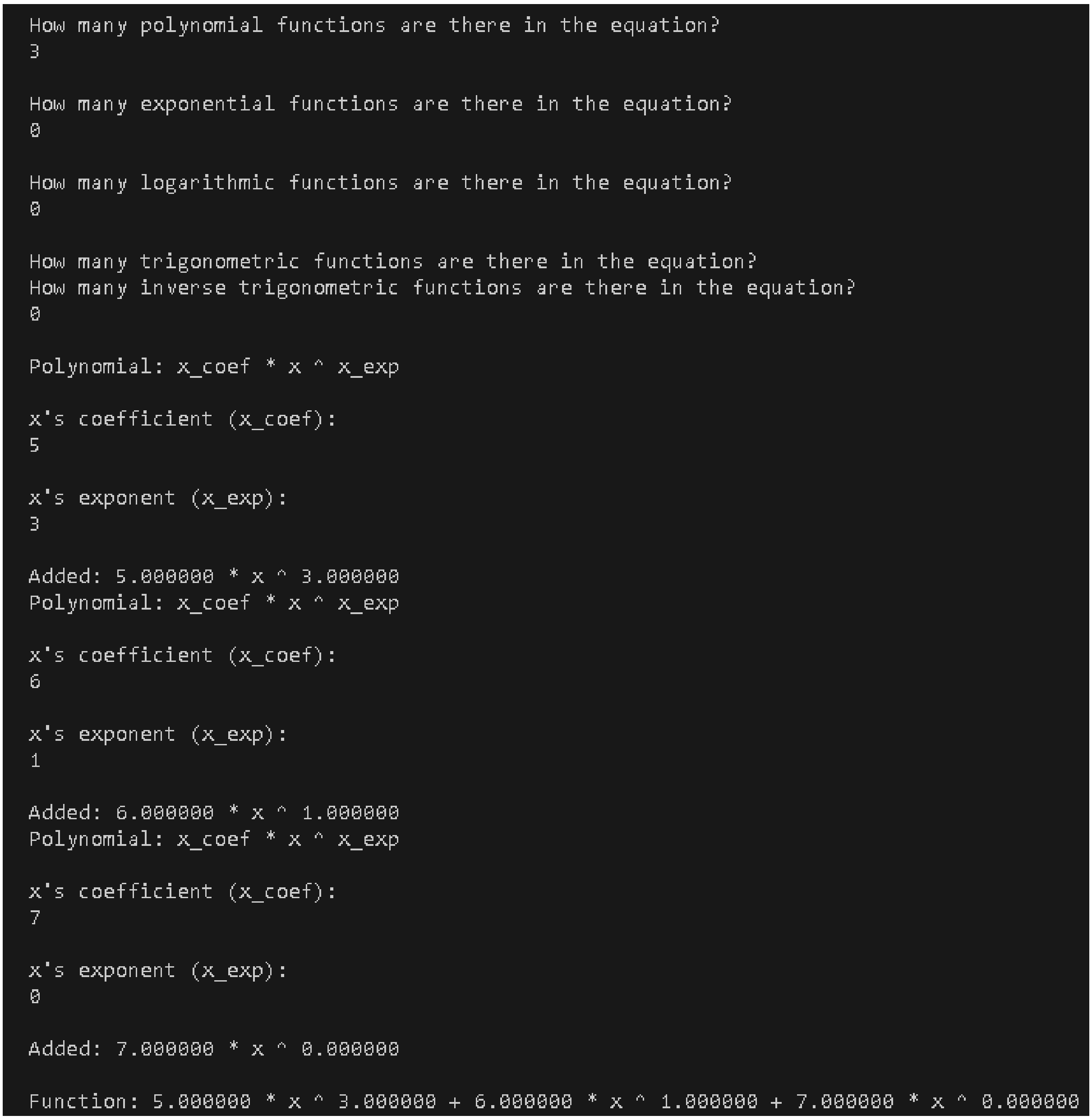
Inverse Trigonometric

$$fn_{coef} \times trig_{fn}\left(x_{coef} \times x^{x_{\mathrm{exp}}}\right)^{fn_{\mathrm{exp}}}$$

$$trig_{fn} = \begin{cases} \text{arcsin} & 0 \\ \text{arccos} & 1 \\ \text{arctan} & 2 \\ \text{arccot} & 3 \end{cases}$$

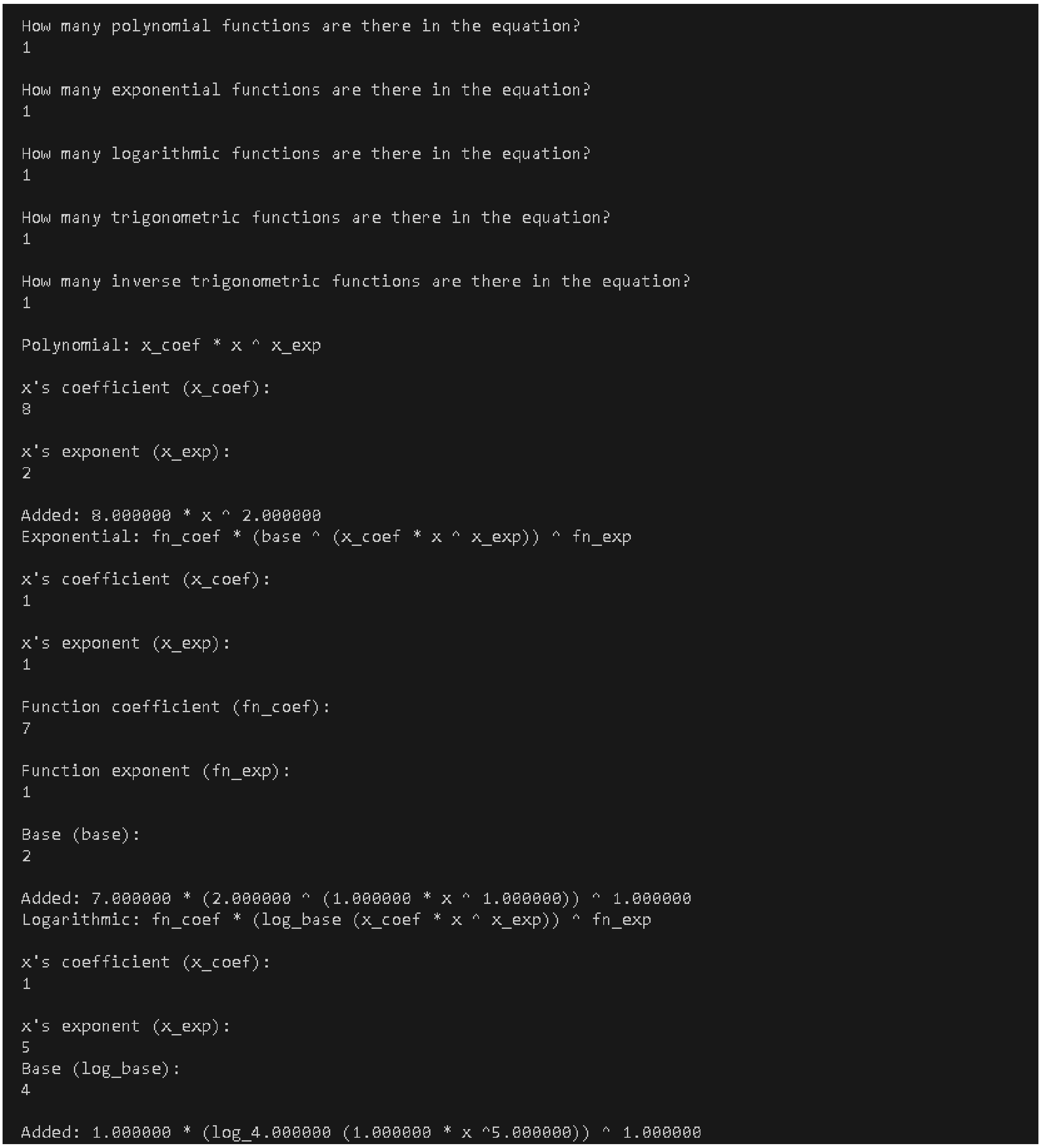
Examples

$$5x^3 + 6x + 7$$

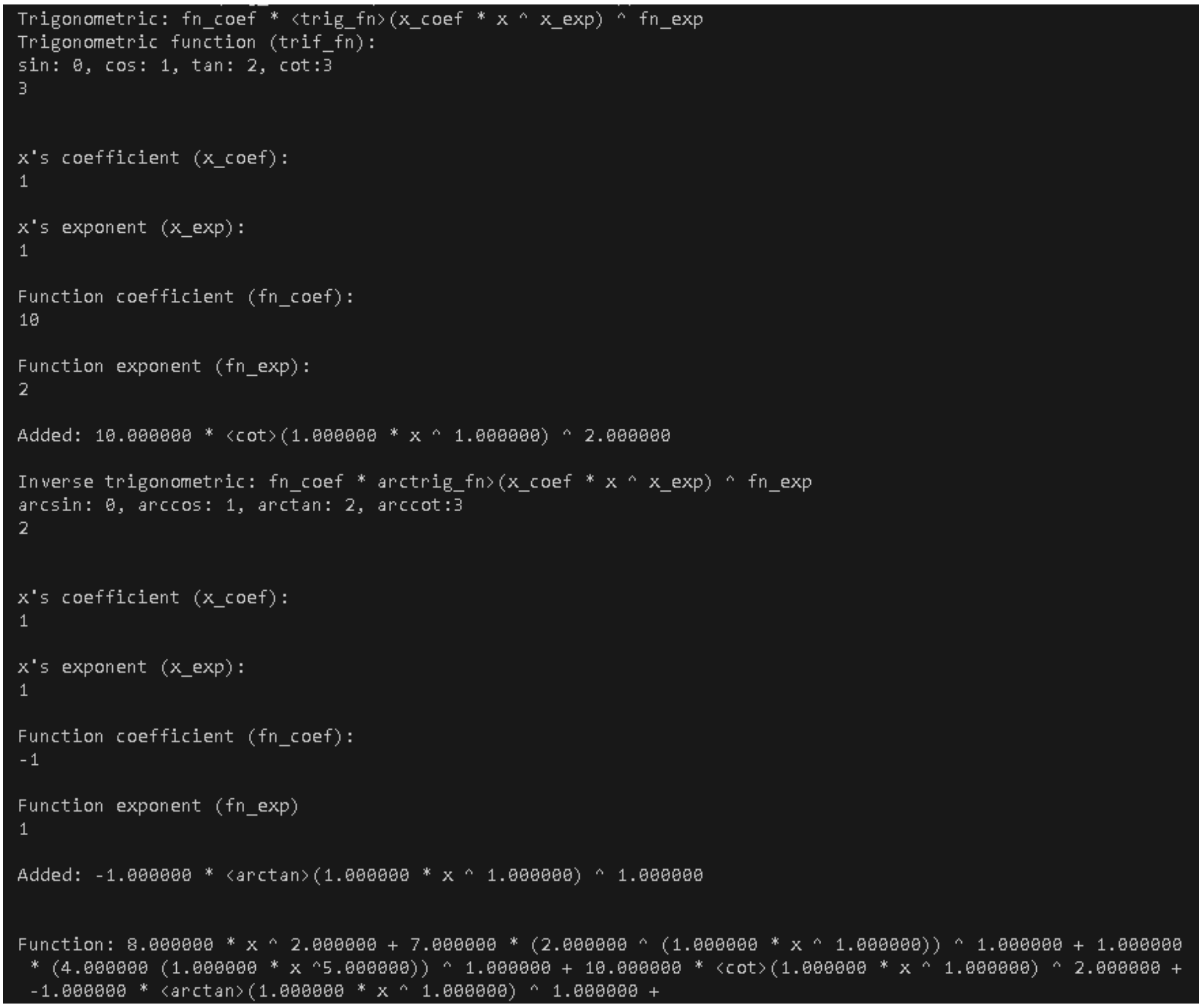


```
How many polynomial functions are there in the equation?
3

How many exponential functions are there in the equation?
0

How many logarithmic functions are there in the equation?
0

How many trigonometric functions are there in the equation?
How many inverse trigonometric functions are there in the equation?
0

Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
5

x's exponent (x_exp):
3

Added: 5.000000 * x ^ 3.000000
Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
6

x's exponent (x_exp):
1

Added: 6.000000 * x ^ 1.000000
Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
7

x's exponent (x_exp):
0

Added: 7.000000 * x ^ 0.000000

Function: 5.000000 * x ^ 3.000000 + 6.000000 * x ^ 1.000000 + 7.000000 * x ^ 0.000000
```

$$8x^2 + 7 \times 2^x - \log_4\left(x^5\right) + 10 \times \cot^2(x) - \arctan(x)$$



```
How many polynomial functions are there in the equation?
1

How many exponential functions are there in the equation?
1

How many logarithmic functions are there in the equation?
1

How many trigonometric functions are there in the equation?
1

How many inverse trigonometric functions are there in the equation?
1

Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
8

x's exponent (x_exp):
2

Added: 8.000000 * x ^ 2.000000
Exponential: fn_coef * (base ^ (x_coef * x ^ x_exp)) ^ fn_exp

x's coefficient (x_coef):
1

x's exponent (x_exp):
1

Function coefficient (fn_coef):
7

Function exponent (fn_exp):
1

Base (base):
2

Added: 7.000000 * (2.000000 ^ (1.000000 * x ^ 1.000000)) ^ 1.000000
Logarithmic: fn_coef * (log_base (x_coef * x ^ x_exp)) ^ fn_exp

x's coefficient (x_coef):
1

x's exponent (x_exp):
5
Base (log_base):
4

Added: 1.000000 * (log_4.000000 (1.000000 * x ^5.000000)) ^ 1.000000
```



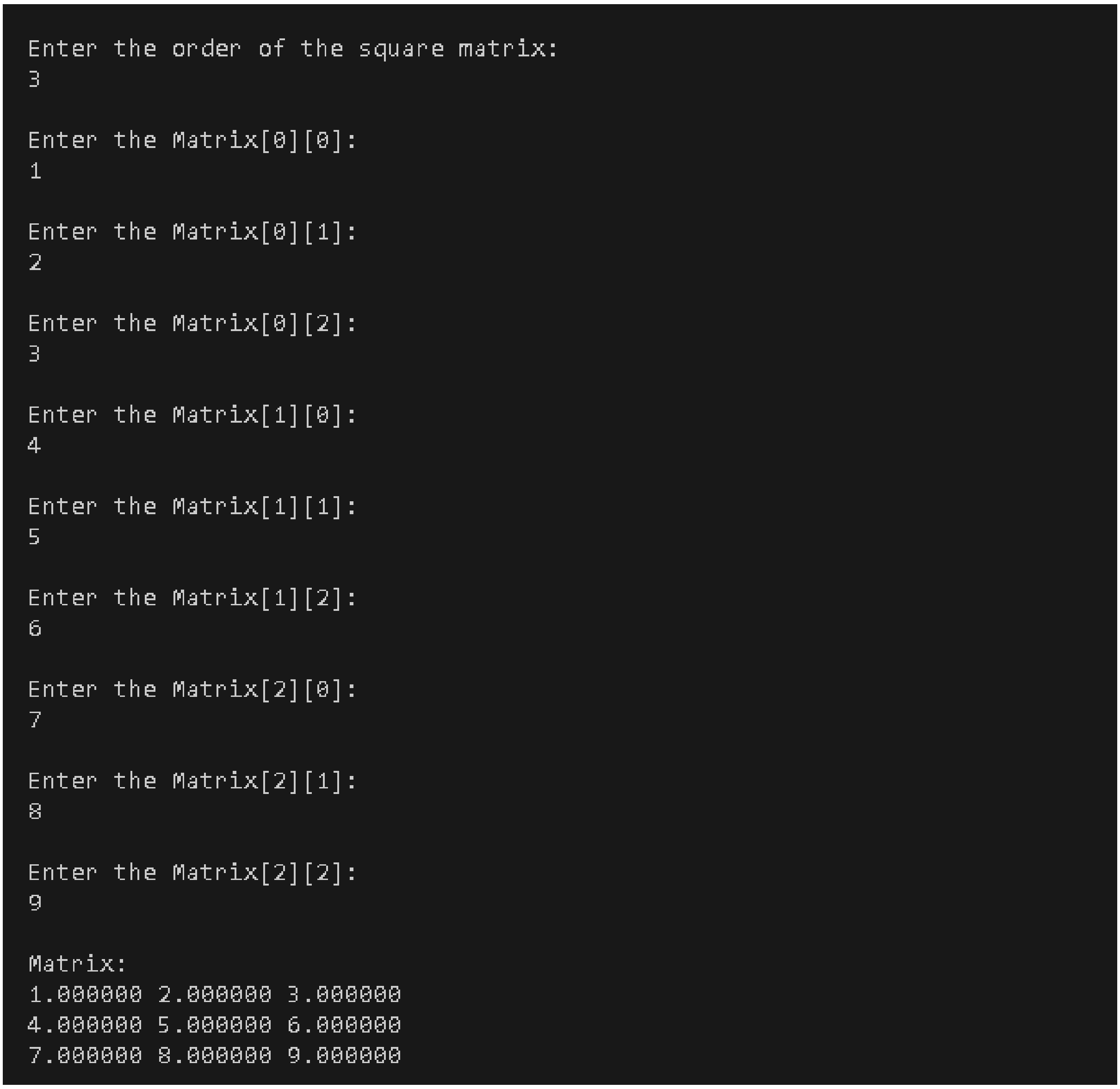
```
Trigonometric: fn_coef * <trig_fn>(x_coef * x ^ x_exp) ^ fn_exp
Trigonometric function (trif_fn):
sin: 0, cos: 1, tan: 2, cot:3
3


x's coefficient (x_coef):
1

x's exponent (x_exp):
1

Function coefficient (fn_coef):
10

Function exponent (fn_exp):
2

Added: 10.000000 * <cot>(1.000000 * x ^ 1.000000) ^ 2.000000

Inverse trigonometric: fn_coef * arctrig_fn>(x_coef * x ^ x_exp) ^ fn_exp
arcsin: 0, arccos: 1, arctan: 2, arccot:3
2


x's coefficient (x_coef):
1

x's exponent (x_exp):
1

Function coefficient (fn_coef):
-1

Function exponent (fn_exp)
1

Added: -1.000000 * <arctan>(1.000000 * x ^ 1.000000) ^ 1.000000


Function: 8.000000 * x ^ 2.000000 + 7.000000 * (2.000000 ^ (1.000000 * x ^ 1.000000)) ^ 1.000000 + 1.000000
 * (4.000000 (1.000000 * x ^5.000000)) ^ 1.000000 + 10.000000 * <cot>(1.000000 * x ^ 1.000000) ^ 2.000000 +
 -1.000000 * <arctan>(1.000000 * x ^ 1.000000) ^ 1.000000 +
```

Matrix Input

For the inverse of the matrix (4) and methods for solving linear equations (5, 6), the first parameter requested is the value of N for an NxN square matrix. After entering this value, the elements of the matrix are taken row by row.

Example

$$N = 3, \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



```
Enter the order of the square matrix:
3

Enter the Matrix[0][0]:
1

Enter the Matrix[0][1]:
2

Enter the Matrix[0][2]:
3

Enter the Matrix[1][0]:
4

Enter the Matrix[1][1]:
5

Enter the Matrix[1][2]:
6

Enter the Matrix[2][0]:
7

Enter the Matrix[2][1]:
8

Enter the Matrix[2][2]:
9

Matrix:
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
7.000000 8.000000 9.000000
```

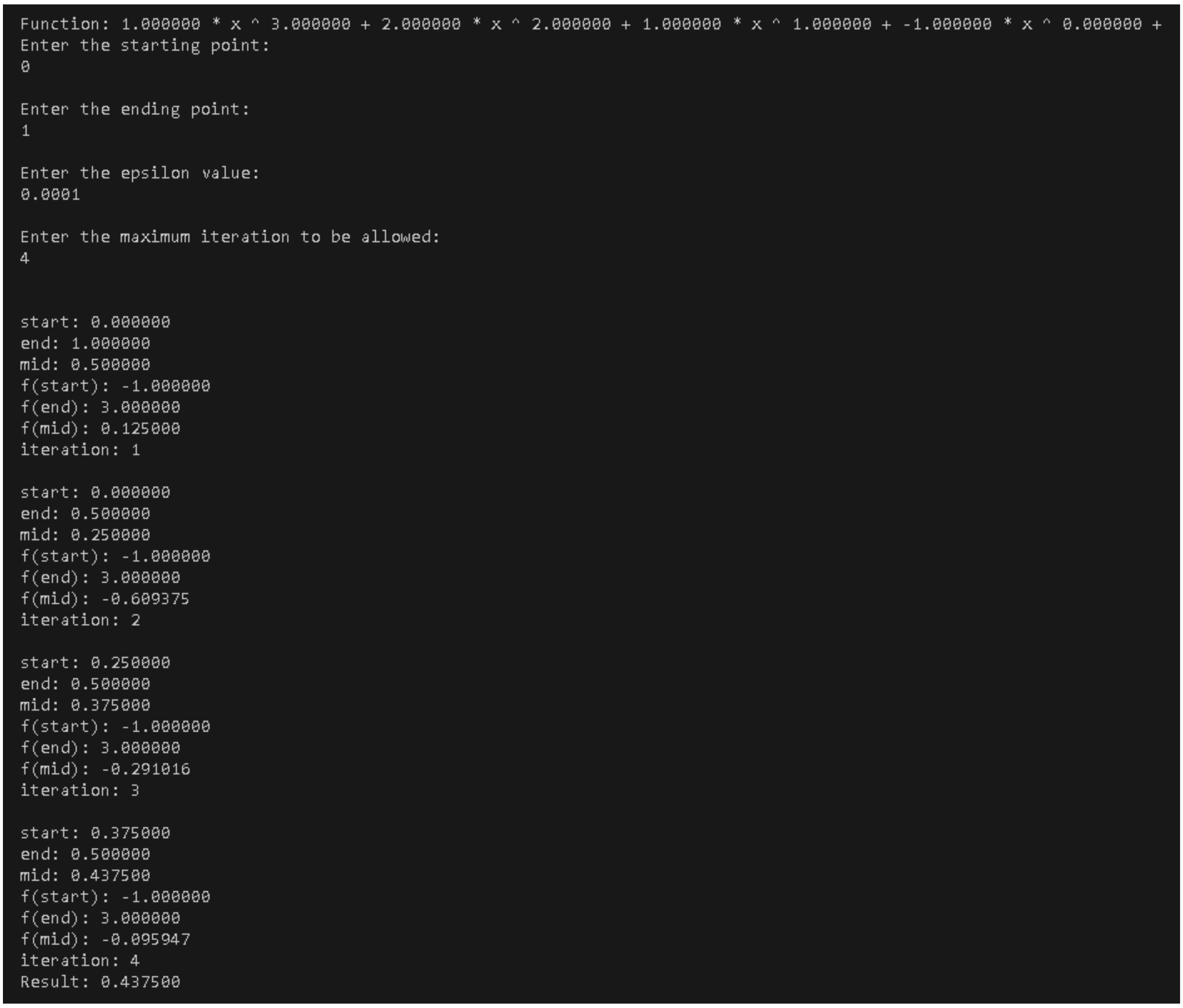
Bisection Method

Parameters

Function
start
end
epsilon
Stopping criterion $= \begin{cases} a - b < \epsilon \\ iteration \geq \max iteration \end{cases}$
Max iterations:

Example

Function $x^3 + 2x^2 + x - 1$
start: *0*
end: *1*
epsilon: *0.0001*
Max iterations: *4*



```
Function: 1.000000 * x ^ 3.000000 + 2.000000 * x ^ 2.000000 + 1.000000 * x ^ 1.000000 + -1.000000 * x ^ 0.000000 +
Enter the starting point:
0

Enter the ending point:
1

Enter the epsilon value:
0.0001

Enter the maximum iteration to be allowed:
4


start: 0.000000
end: 1.000000
mid: 0.500000
f(start): -1.000000
f(end): 3.000000
f(mid): 0.125000
iteration: 1

start: 0.000000
end: 0.500000
mid: 0.250000
f(start): -1.000000
f(end): 3.000000
f(mid): -0.609375
iteration: 2

start: 0.250000
end: 0.500000
mid: 0.375000
f(start): -1.000000
f(end): 3.000000
f(mid): -0.291016
iteration: 3

start: 0.375000
end: 0.500000
mid: 0.437500
f(start): -1.000000
f(end): 3.000000
f(mid): -0.095947
iteration: 4
Result: 0.437500
```

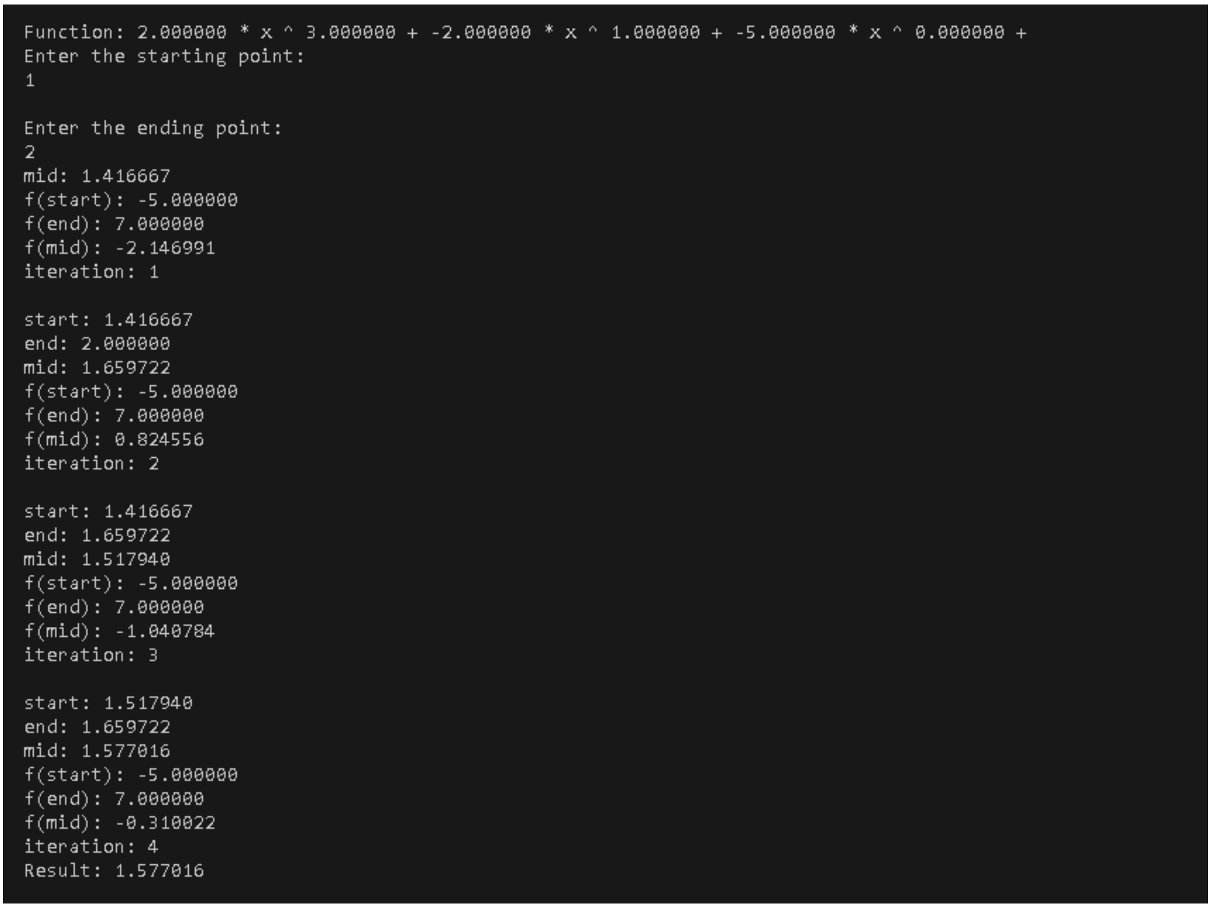
Regula-Falsi

Parameters

Function
start
end
epsilon
Stopping criterion: $= x_{n+1} - x_n < \epsilon$
Max iterations

Example

Function $2x^3 - 2x - 5$
start: *1*
end: *2*
epsilon: *0.0001*
Max iterations: *4*



```
Function: 2.000000 * x ^ 3.000000 + -2.000000 * x ^ 1.000000 + -5.000000 * x ^ 0.000000 +
Enter the starting point:
1

Enter the ending point:
2
mid: 1.416667
f(start): -5.000000
f(end): 7.000000
f(mid): -2.146991
iteration: 1

start: 1.416667
end: 2.000000
mid: 1.659722
f(start): -5.000000
f(end): 7.000000
f(mid): 0.824556
iteration: 2

start: 1.416667
end: 1.659722
mid: 1.517940
f(start): -5.000000
f(end): 7.000000
f(mid): -1.040784
iteration: 3

start: 1.517940
end: 1.659722
mid: 1.577016
f(start): -5.000000
f(end): 7.000000
f(mid): -0.310022
iteration: 4
Result: 1.577016
```

Newton-Raphson

Parameters

Fonksiyon

x_0***:*** *Start value*

epsilon

Stopping criterion$= x_{n+1} - x_n < \epsilon$

Max iterations

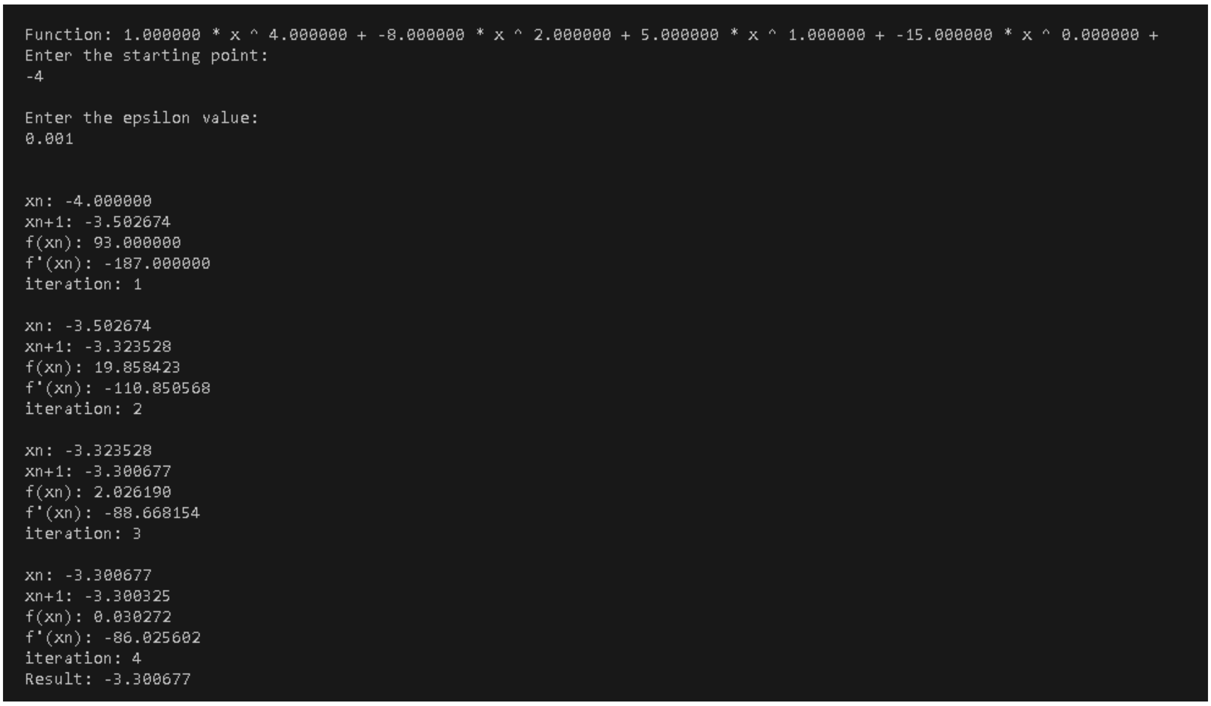
Example

Function $x^4 - 8x^2 + 5x - 15$

x_0***:*** *-4*

epsilon: *0.001*

Max iterations: *100(If the iteration reaches this value, it means the method is diverging.)*



```
Function: 1.000000 * x ^ 4.000000 + -8.000000 * x ^ 2.000000 + 5.000000 * x ^ 1.000000 + -15.000000 * x ^ 0.000000 +
Enter the starting point:
-4

Enter the epsilon value:
0.001


xn: -4.000000
xn+1: -3.502674
f(xn): 93.000000
f'(xn): -187.000000
iteration: 1

xn: -3.502674
xn+1: -3.323528
f(xn): 19.858423
f'(xn): -110.850568
iteration: 2

xn: -3.323528
xn+1: -3.300677
f(xn): 2.026190
f'(xn): -88.668154
iteration: 3

xn: -3.300677
xn+1: -3.300325
f(xn): 0.030272
f'(xn): -86.025602
iteration: 4
Result: -3.300677
```

Matrix Inverse

Parameters

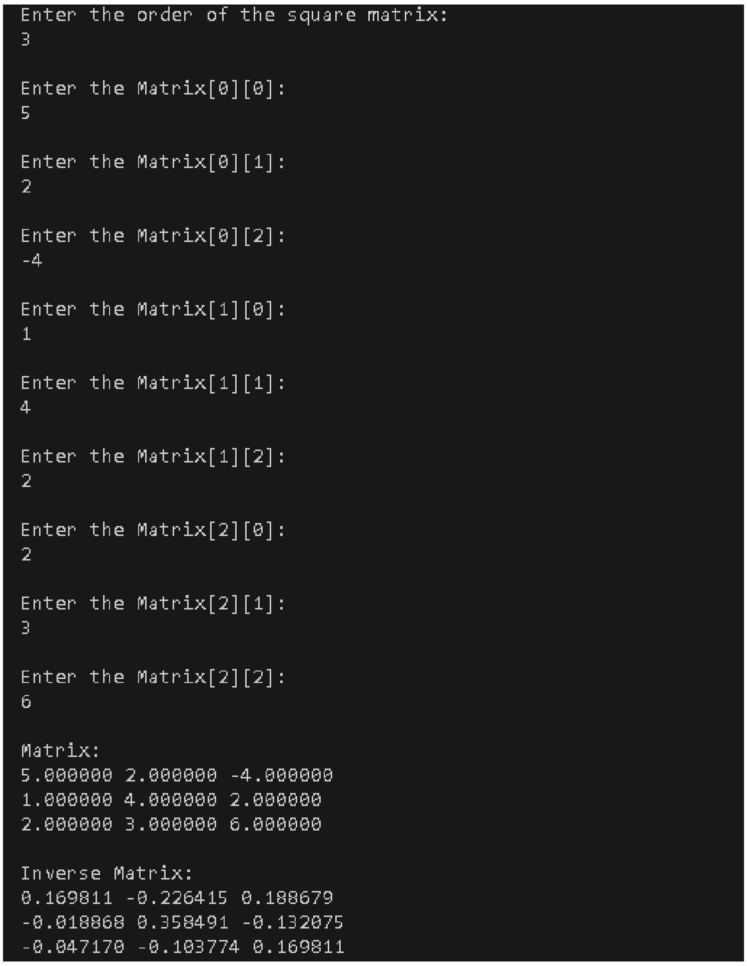
***N:** Order of a matrix*

Matrix: $$Matrix = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Example

***N:** 3*

Matrix: $$Matrix = \begin{bmatrix} 5 & 2 & -4 \\ 1 & 4 & 2 \\ 2 & 3 & 6 \end{bmatrix}$$



```
Enter the order of the square matrix:
3

Enter the Matrix[0][0]:
5

Enter the Matrix[0][1]:
2

Enter the Matrix[0][2]:
-4

Enter the Matrix[1][0]:
1

Enter the Matrix[1][1]:
4

Enter the Matrix[1][2]:
2

Enter the Matrix[2][0]:
2

Enter the Matrix[2][1]:
3

Enter the Matrix[2][2]:
6

Matrix:
5.000000 2.000000 -4.000000
1.000000 4.000000 2.000000
2.000000 3.000000 6.000000

Inverse Matrix:
0.169811 -0.226415 0.188679
-0.018868 0.358491 -0.132075
-0.047170 -0.103774 0.169811
```

Gauss Elimination

Parameters

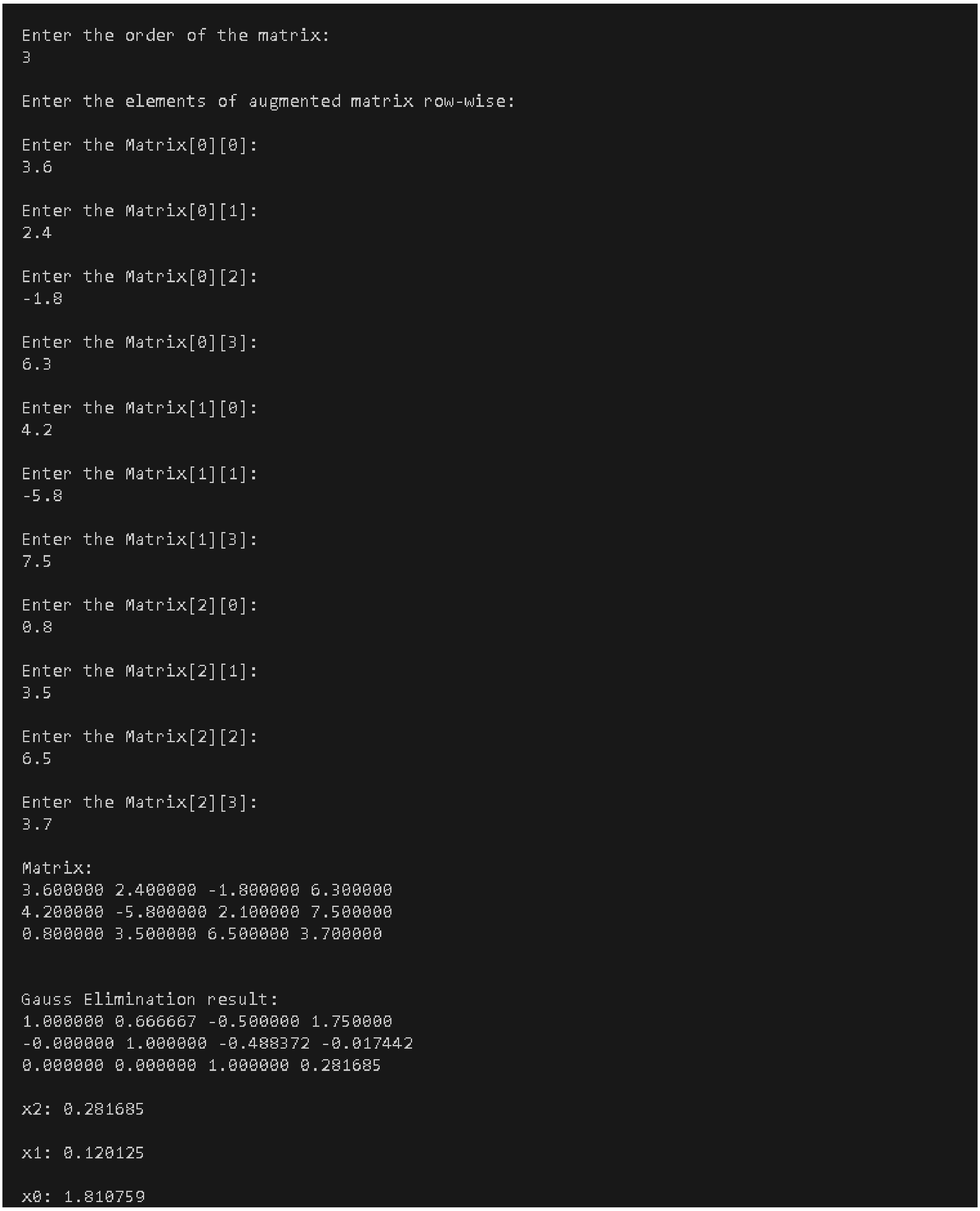
N: *Order of a matrix*

Matris: $Matrix = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$

Example

N: *3*

Matrix:$Matrix = \begin{bmatrix} 3.6 & 2.4 & -1.8 \\ 4.2 & -5.8 & 2.1 \\ 0.8 & 3.5 & 6.5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6.3 \\ 7.5 \\ 3.7 \end{bmatrix}$



```
Enter the order of the matrix:
3

Enter the elements of augmented matrix row-wise:

Enter the Matrix[0][0]:
3.6

Enter the Matrix[0][1]:
2.4

Enter the Matrix[0][2]:
-1.8

Enter the Matrix[0][3]:
6.3

Enter the Matrix[1][0]:
4.2

Enter the Matrix[1][1]:
-5.8

Enter the Matrix[1][3]:
7.5

Enter the Matrix[2][0]:
0.8

Enter the Matrix[2][1]:
3.5

Enter the Matrix[2][2]:
6.5

Enter the Matrix[2][3]:
3.7

Matrix:
3.600000 2.400000 -1.800000 6.300000
4.200000 -5.800000 2.100000 7.500000
0.800000 3.500000 6.500000 3.700000


Gauss Elimination result:
1.000000 0.666667 -0.500000 1.750000
-0.000000 1.000000 -0.488372 -0.017442
0.000000 0.000000 1.000000 0.281685

x2: 0.281685

x1: 0.120125

x0: 1.810759
```

Gauss-Seidel

Parameters

N: *Order of a matrix*

Matrix: $Matrix = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$

epsilon

Stopping criterion: $\triangle x,\ \triangle y,\ \triangle z\ <\ \epsilon$

Max iterations

Örnek

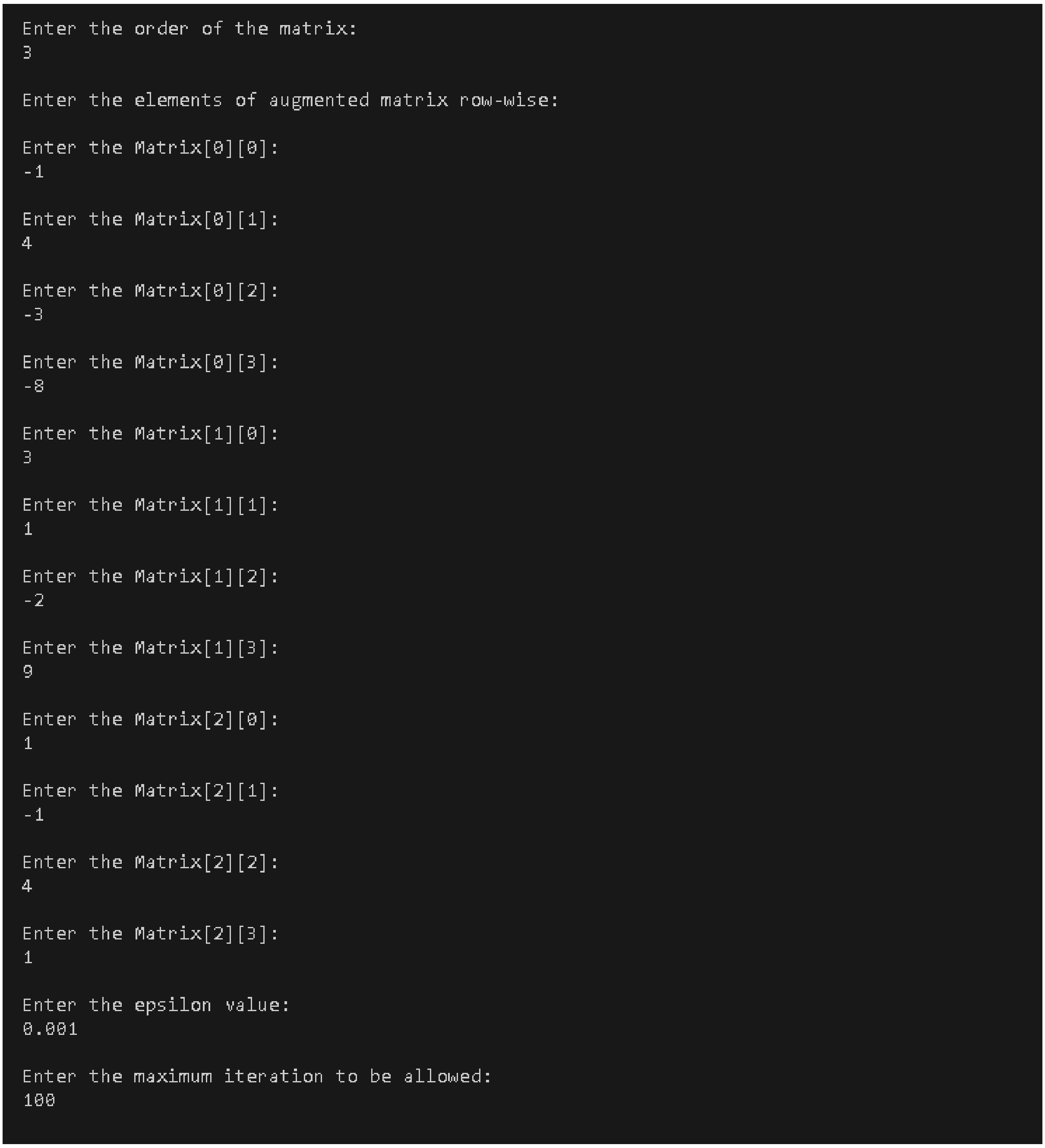
N: *3*

Matrix:$Matrix = \begin{bmatrix} -1 & 4 & -3 \\ 3 & 1 & -2 \\ 1 & -1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -8 \\ 9 \\ 1 \end{bmatrix}$

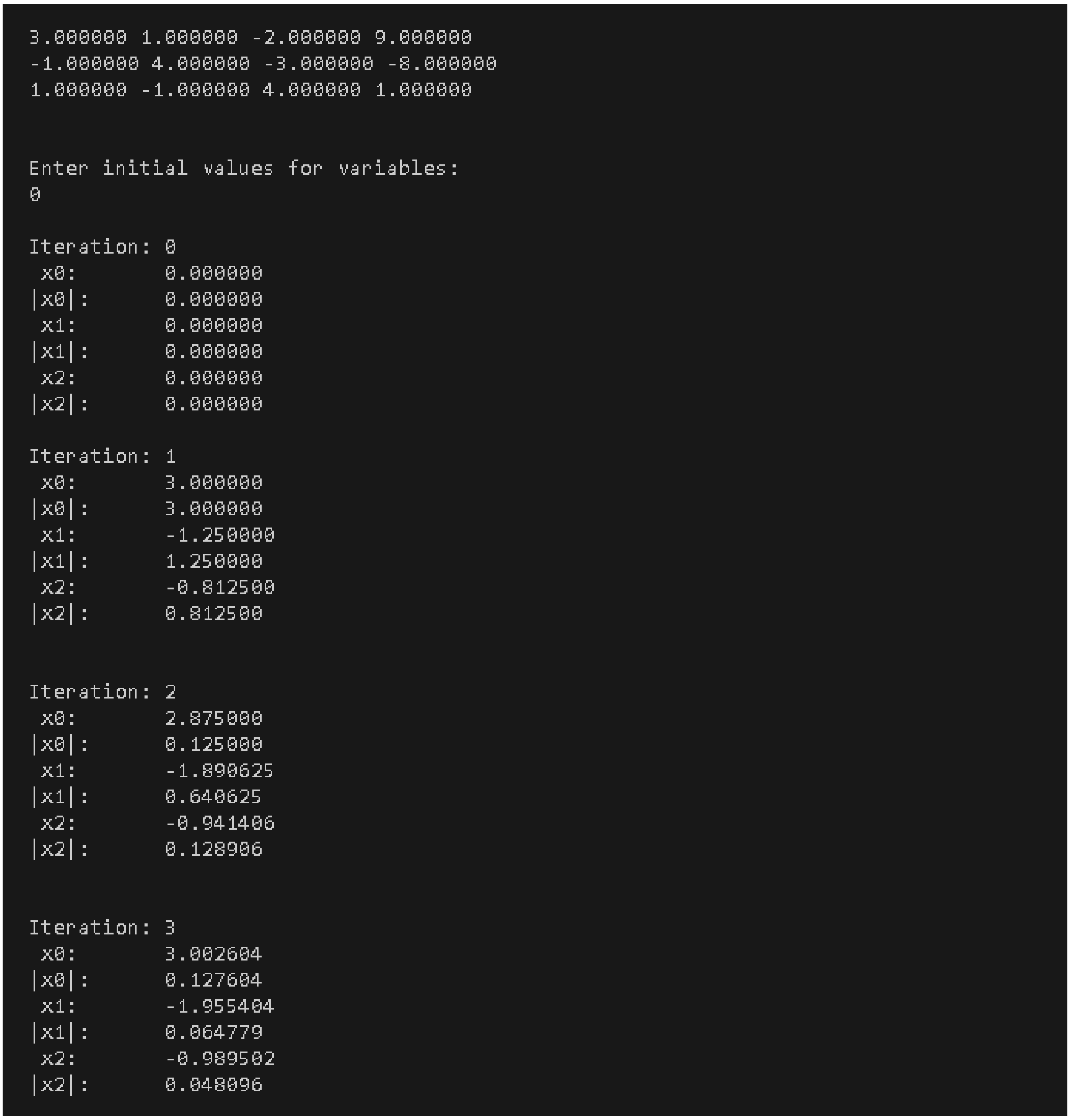
epsilon: *0.001*

Stopping criterion: $\triangle x,\ \triangle y,\ \triangle z\ <\ 0.001$

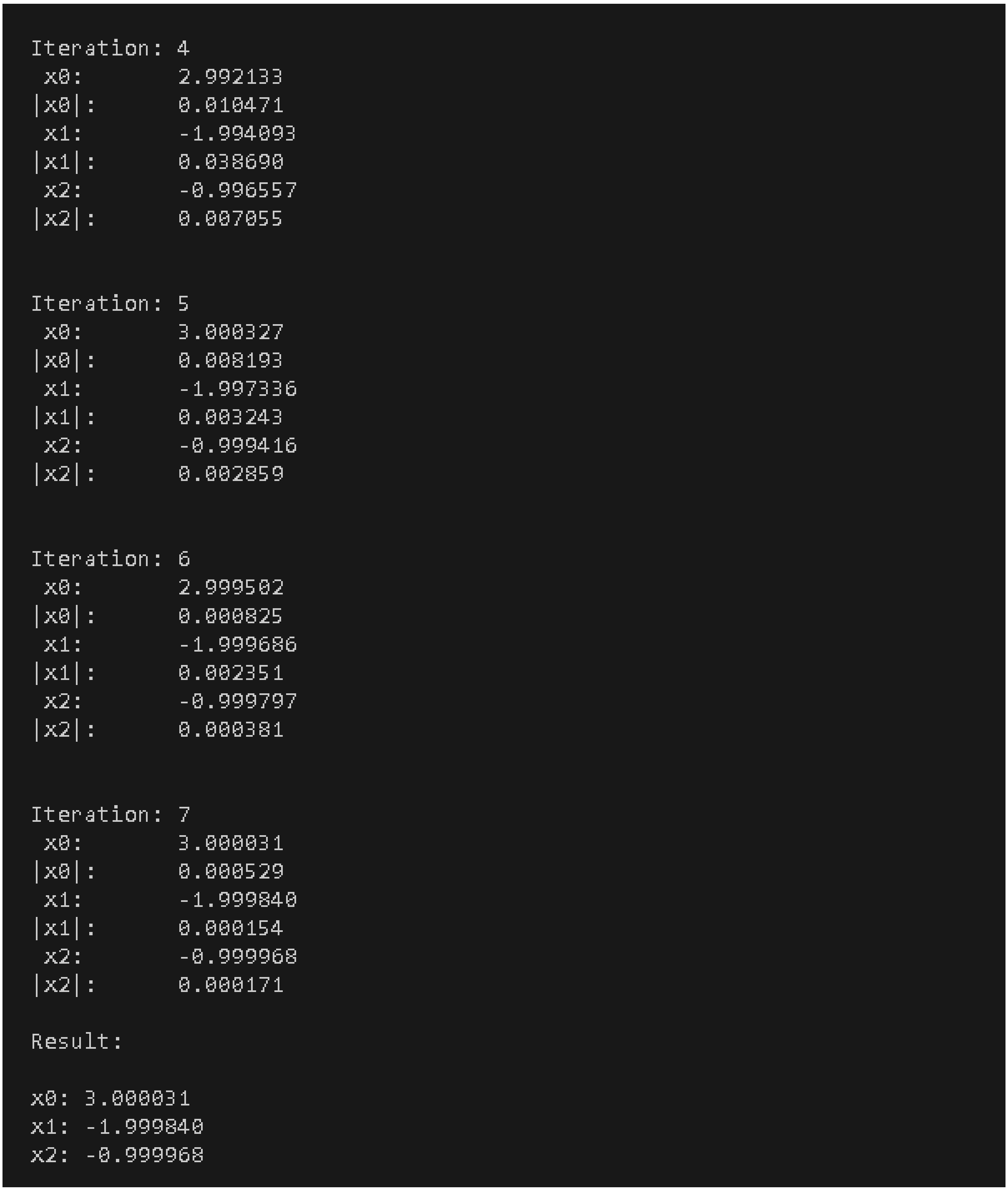
Max iterations: *100*



```
Enter the order of the matrix:
3

Enter the elements of augmented matrix row-wise:

Enter the Matrix[0][0]:
-1

Enter the Matrix[0][1]:
4

Enter the Matrix[0][2]:
-3

Enter the Matrix[0][3]:
-8

Enter the Matrix[1][0]:
3

Enter the Matrix[1][1]:
1

Enter the Matrix[1][2]:
-2

Enter the Matrix[1][3]:
9

Enter the Matrix[2][0]:
1

Enter the Matrix[2][1]:
-1

Enter the Matrix[2][2]:
4

Enter the Matrix[2][3]:
1

Enter the epsilon value:
0.001

Enter the maximum iteration to be allowed:
100
```



```
3.000000 1.000000 -2.000000 9.000000
-1.000000 4.000000 -3.000000 -8.000000
1.000000 -1.000000 4.000000 1.000000


Enter initial values for variables:
0

Iteration: 0
 x0:       0.000000
|x0|:      0.000000
 x1:       0.000000
|x1|:      0.000000
 x2:       0.000000
|x2|:      0.000000

Iteration: 1
 x0:       3.000000
|x0|:      3.000000
 x1:       -1.250000
|x1|:      1.250000
 x2:       -0.812500
|x2|:      0.812500


Iteration: 2
 x0:       2.875000
|x0|:      0.125000
 x1:       -1.890625
|x1|:      0.640625
 x2:       -0.941406
|x2|:      0.128906


Iteration: 3
 x0:       3.002604
|x0|:      0.127604
 x1:       -1.955404
|x1|:      0.064779
 x2:       -0.989502
|x2|:      0.048096
```



```
Iteration: 4
 x0:       2.992133
|x0|:      0.010471
 x1:       -1.994093
|x1|:      0.038690
 x2:       -0.996557
|x2|:      0.007055


Iteration: 5
 x0:       3.000327
|x0|:      0.008193
 x1:       -1.997336
|x1|:      0.003243
 x2:       -0.999416
|x2|:      0.002859


Iteration: 6
 x0:       2.999502
|x0|:      0.000825
 x1:       -1.999686
|x1|:      0.002351
 x2:       -0.999797
|x2|:      0.000381


Iteration: 7
 x0:       3.000031
|x0|:      0.000529
 x1:       -1.999840
|x1|:      0.000154
 x2:       -0.999968
|x2|:      0.000171

Result:

x0: 3.000031
x1: -1.999840
x2: -0.999968
```

Numerical Differentiation

Parameters

function

numerical differentiation method: *difference, central difference forward difference, backward*

h: *The difference value for the derivative operation.*

x

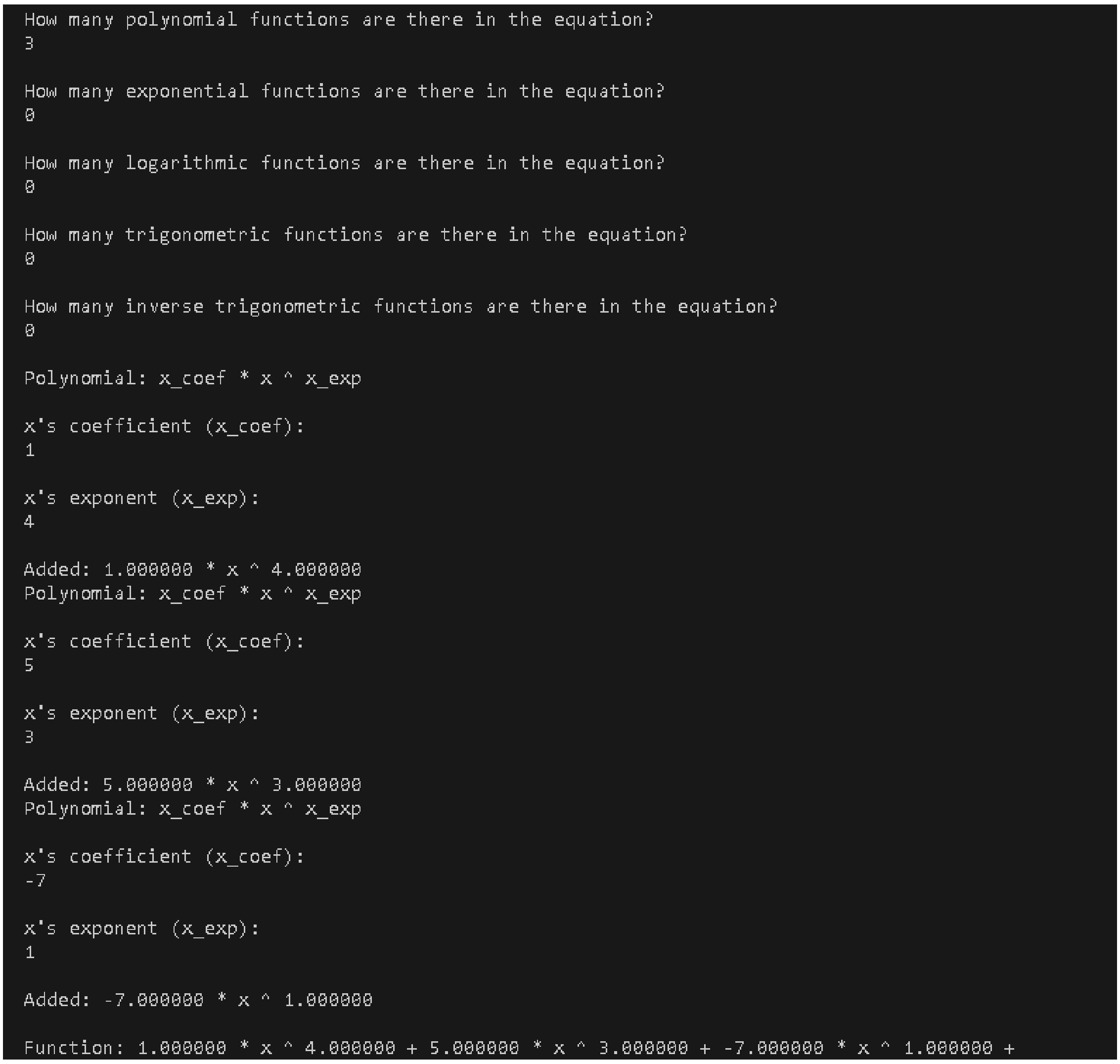
Example

funtion: $x^4 + 5x^3 - 7$

numerical differentiation method: *Geri fark, İleri fark, Merkezi fark*

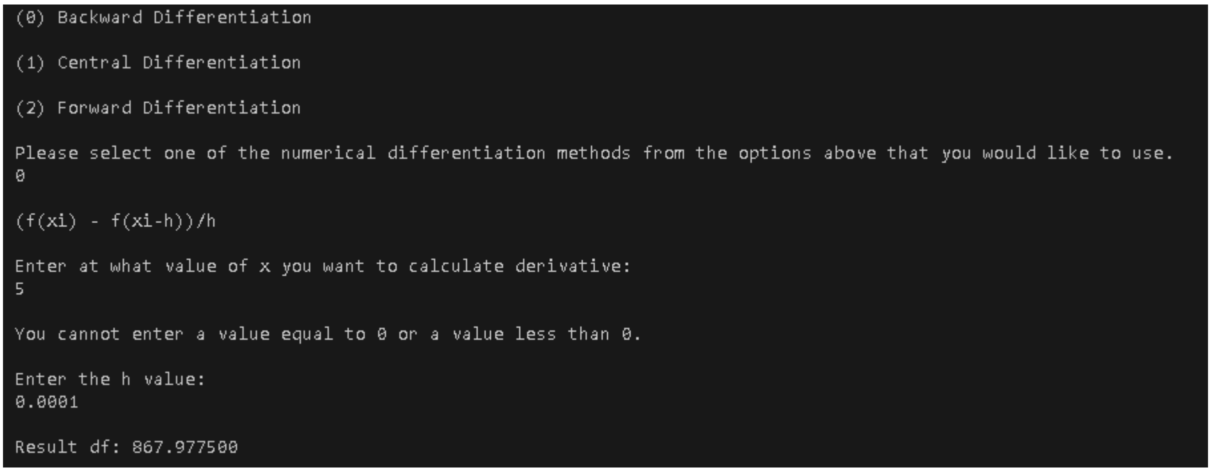
h: *türev işlemi için fark değeri*

x: *5*



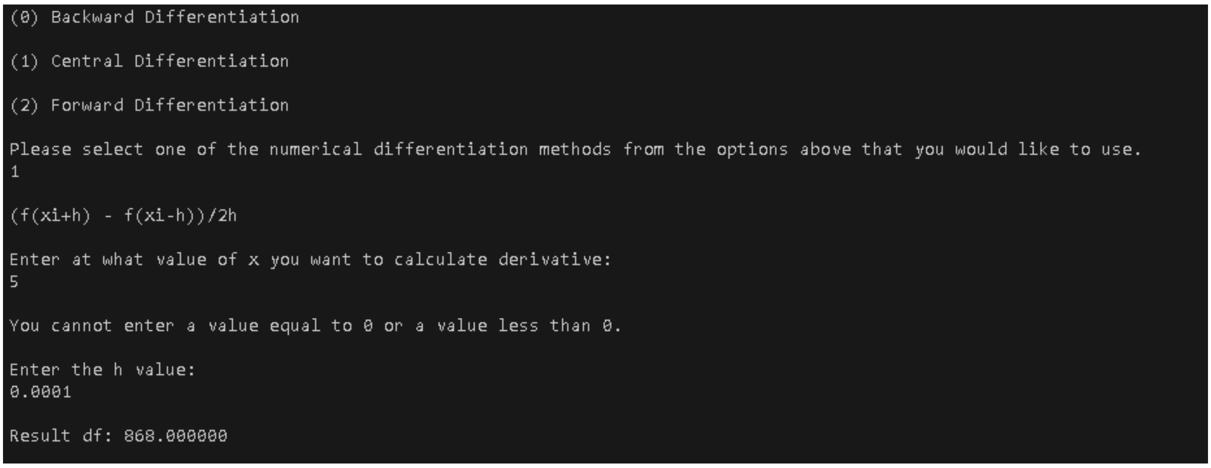
```
How many polynomial functions are there in the equation?
3

How many exponential functions are there in the equation?
0

How many logarithmic functions are there in the equation?
0

How many trigonometric functions are there in the equation?
0

How many inverse trigonometric functions are there in the equation?
0

Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
1

x's exponent (x_exp):
4

Added: 1.000000 * x ^ 4.000000
Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
5

x's exponent (x_exp):
3

Added: 5.000000 * x ^ 3.000000
Polynomial: x_coef * x ^ x_exp

x's coefficient (x_coef):
-7

x's exponent (x_exp):
1

Added: -7.000000 * x ^ 1.000000

Function: 1.000000 * x ^ 4.000000 + 5.000000 * x ^ 3.000000 + -7.000000 * x ^ 1.000000 +
```

Backward Difference Example



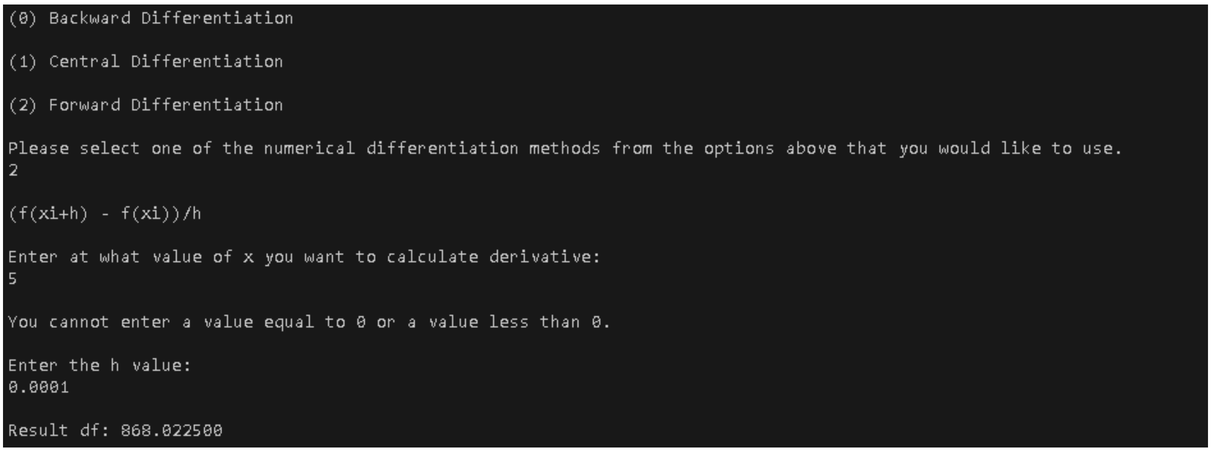
```
(0) Backward Differentiation

(1) Central Differentiation

(2) Forward Differentiation

Please select one of the numerical differentiation methods from the options above that you would like to use.
0

(f(xi) - f(xi-h))/h

Enter at what value of x you want to calculate derivative:
5

You cannot enter a value equal to 0 or a value less than 0.

Enter the h value:
0.0001

Result df: 867.977500
```

Central Difference Example



```
(0) Backward Differentiation

(1) Central Differentiation

(2) Forward Differentiation

Please select one of the numerical differentiation methods from the options above that you would like to use.
1

(f(xi+h) - f(xi-h))/2h

Enter at what value of x you want to calculate derivative:
5

You cannot enter a value equal to 0 or a value less than 0.

Enter the h value:
0.0001

Result df: 868.000000
```

Forward Difference Example



```
(0) Backward Differentiation

(1) Central Differentiation

(2) Forward Differentiation

Please select one of the numerical differentiation methods from the options above that you would like to use.
2

(f(xi+h) - f(xi))/h

Enter at what value of x you want to calculate derivative:
5

You cannot enter a value equal to 0 or a value less than 0.

Enter the h value:
0.0001

Result df: 868.022500
```

Simpson Method

Parameters

function

lower limit: $\int_a^b f(x)\,dx - > \ a\ değeri$

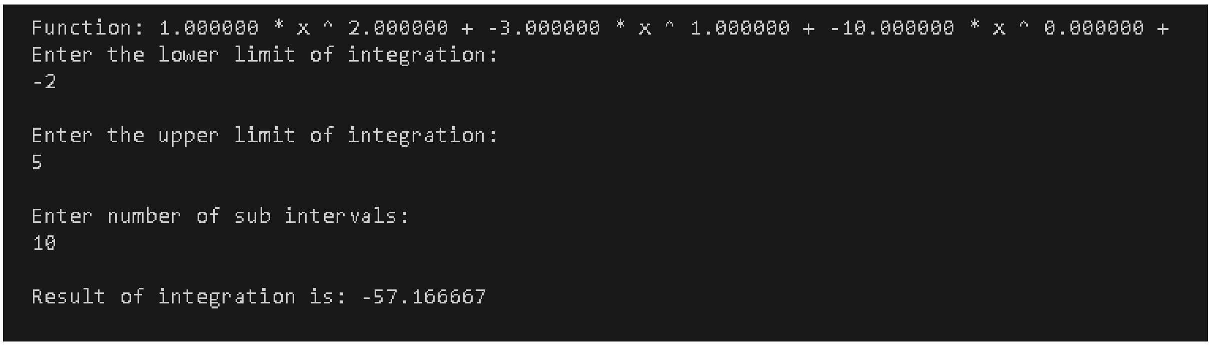
$$\int_a^b f(x)\,dx - > \ b\ değeri$$

upper limit:

number of subintervals: *simpson yöntemi için altaralık sayısı(arttıkça gerçek değere yakınsanır)*

Example

function: $x^2 - 3x - 10$
lower limit: *-2*
upper limit: *5*
number of subintervals: *10*



```
Function: 1.000000 * x ^ 2.000000 + -3.000000 * x ^ 1.000000 + -10.000000 * x ^ 0.000000 +
Enter the lower limit of integration:
-2

Enter the upper limit of integration:
5

Enter number of sub intervals:
10

Result of integration is: -57.166667
```

Trapez Method

Parameters

function:

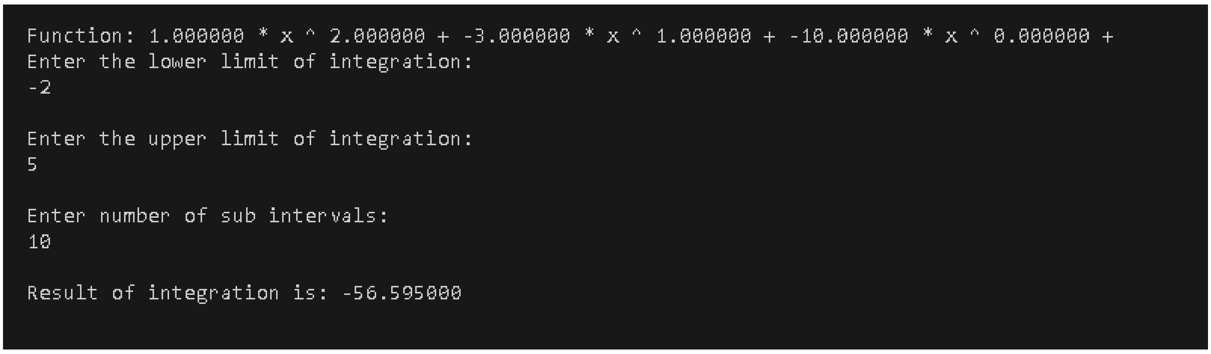
lower limit: $\int_a^b f(x)\,dx\ - >\ a\ de\breve{g}eri$

upper limit: $\int_a^b f(x)\,dx\ - >\ b\ de\breve{g}eri$

number of subintervals: *The number of intervals for the trapezoidal method (increasing leads to closer approximation to the true value)*

Example

function: $x^2 - 3x - 10$
lower limit: *-2*
upper limit: *5*
number of subintervals: *10*



```
Function: 1.000000 * x ^ 2.000000 + -3.000000 * x ^ 1.000000 + -10.000000 * x ^ 0.000000 +
Enter the lower limit of integration:
-2

Enter the upper limit of integration:
5

Enter number of sub intervals:
10

Result of integration is: -56.595000
```

Gregory-Newton Interpolation

Parameters

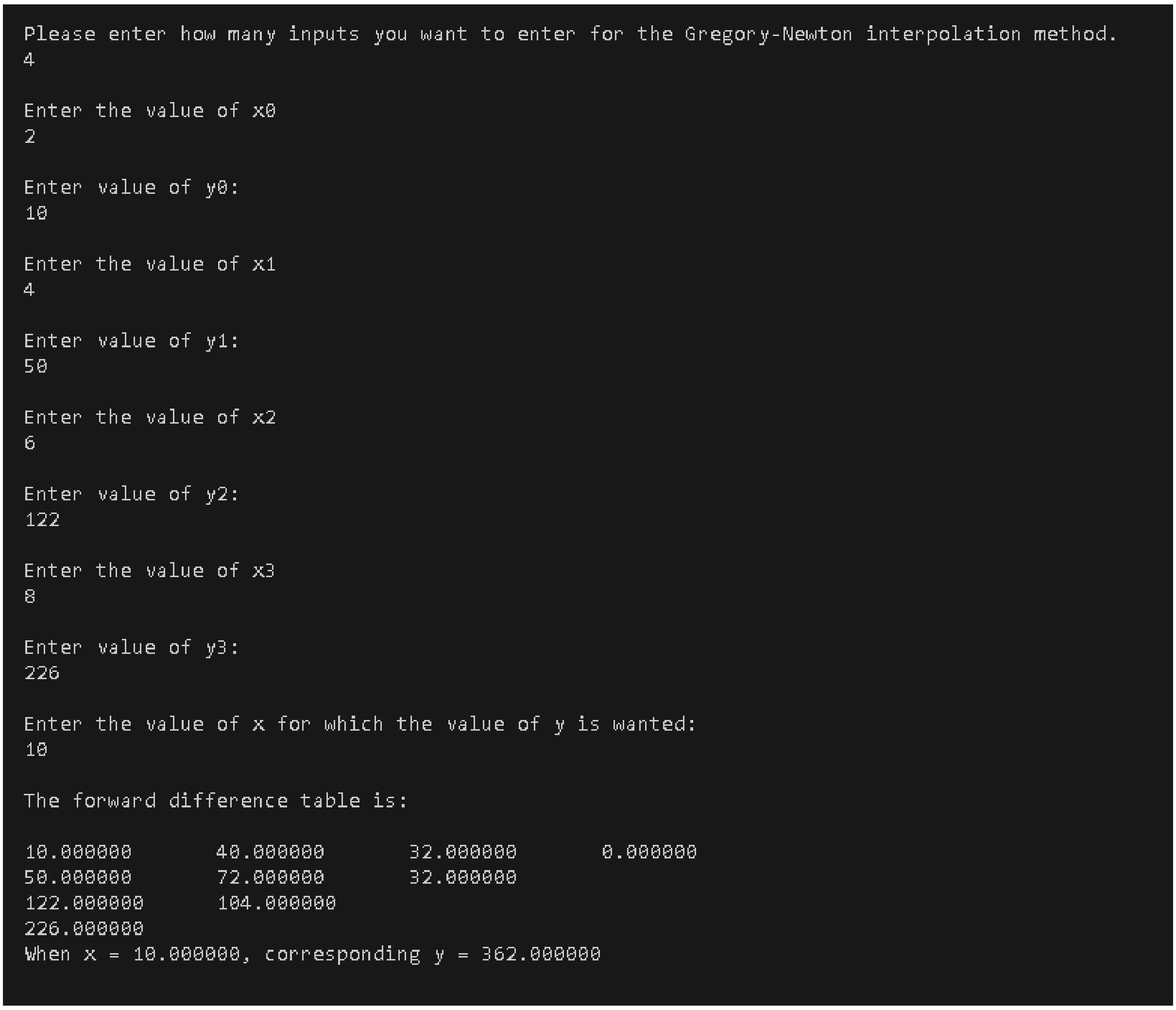
number of inputs
input values: *x values corresponding to y values*
x

Example

number of inputs: *4*

input values: $\begin{bmatrix} x & y \\ 2 & 10 \\ 4 & 50 \\ 6 & 122 \\ 8 & 226 \end{bmatrix}$

x: *10*



```
Please enter how many inputs you want to enter for the Gregory-Newton interpolation method.
4

Enter the value of x0
2

Enter value of y0:
10

Enter the value of x1
4

Enter value of y1:
50

Enter the value of x2
6

Enter value of y2:
122

Enter the value of x3
8

Enter value of y3:
226

Enter the value of x for which the value of y is wanted:
10

The forward difference table is:

10.000000       40.000000       32.000000       0.000000
50.000000       72.000000       32.000000
122.000000      104.000000
226.000000
When x = 10.000000, corresponding y = 362.000000
```