SPL-1 Project Report, 2023

EQUATION MASTERMIND

SE 305 : Software Project lab 1

Submitted by

Md Jihad Hossain

BSSE Roll No. : 1413

BSSE Session:2021-22

Supervised by

Dr.Md.Shariful Islam

Designation: professor

Institute of Information Technology

Institute of Information Technology

University of Dhaka

[14-12-2023]

Prepared by:

Md Jihad Hossain

BSSE 14th batch

Roll -1413

Supervised by:

Dr.Md.Shariful Islam
professor

Institute of Information Technology,

University of Dhaka.

Submission Date : 14 th December, 2023

Supervisor's Approval:

(Signature)

Table of Contents

1. Introduction

"Equation Mastermind" solves mathematical equations & show solution process. In this project,I made an equation solver that can solve two kinds of equations. They are,

(i) Linear Equation
(ii) Polynomial Equation

These types of equations are commonly used in our everyday life. It takes input from the user and solves the equation using some algorithm and prints the result and also plot graph. It not only shows the solution of equations but also shows **step by step solution process**. This program takes input from the I/O console and prints it on the I/O console. This program takes input equations from users in any format. It can help students find the root of these equations and also help to know how to solve them.

1.1 Background Study

To understand this project, some prior study was necessary. These topics are included here.

1.1.1 Determinant of Matrix

The determinant is defined as a scalar value that is associated with the square matrix. We find to determinant when the matrix is only square. We use the *row reduction method* to find the determinant of a matrix, meaning all values of lower triangular become 0. Then multiplication of the diagonal elements of a matrix, it finds the determined value.

1.1.2 Cramer's rule

Cramer's rule is a specific formula used for solving a system of linear equations containing as many equations as unknowns, efficient whenever the system of the linear equation has a unique solution. The solution obtained using Cramer's rule will be in terms of the determinants of the coefficient matrix and matrixes obtained from it by replacing one column with the column vector of the right-hand sides of the equations. Now, let's take an example to illustrate Cramer's Rule. Consider the following system of equations:

$$2x + y = 8$$
$$x - 3y = 1$$

Step 1: Compute the determinant of the coefficient matrix D:

D = | 2 1 |
| 1 -3 |

D = (2 *(-3)) - (1 * 1) = -6 - 1 = -7

Step 2: Compute the determinants D1 and D2:

D1 = | 8 1 |
| 1 -3 |

D2 = | 2 8 |
| 1 1 |

D1 = (8 * (-3)) - (1 * 1) = -24 - 1 = -25
D2 = (2 * 1) - (8 * 1) = 2 - 8 = -6

Step 3: Calculate the values of x and y using the ratios of determinants:

x = D1 / D = (-25) / (-7) = 25/7 ≈ 3.57
y = D2 / D = (-6) / (-7) = 6/7 ≈ 0.86

1.1.3 Bairstow Method

Bairstow method is an efficient algorithm for finding the roots of a real polynomial of arbitrary degree. It uses Newton's method to adjust the coefficients u and v in the quadratic until its roots are also roots of the polynomial being solved. The roots of the quadratic may then be determined and the polynomial may be divided by the quadratic to eliminate those roots. This process is iterated until the polynomial becomes quadratic or linear, and all the roots have been determined.

- The steps involved in the Bairstow method are as follows:

P(x) = a[n] * x^n + a[n-1] * x^(n-1) + ... + a[1] * x + a[0]

- Make initial guesses for the roots. These guesses can be obtained using other root-finding techniques or heuristics.

- Assume the polynomial can be factored into quadratic factors:

P(x) = (x^2 + b[1] * x + b[0]) * (x^2 + c[1] * x + c[0]) * ...

- Equate the coefficients of corresponding powers of x in the above equation to the coefficients of the original polynomial. This will result in a set of simultaneous equations.

- Solve the simultaneous equations to find the values of b and c.

- Update the root approximations using the values of b and c.

- Repeat steps 3-6 until the root approximations converge to the desired accuracy. 4

1.1.4 Polynomial Expressions:

In mathematics a polynomial function is an expression consisting of variables and coefficients, that involves the operations of addition, subtraction, multiplication and non-negative integer exponentiation of variables. An example of a polynomial of single indeterminate x is $^2 - 5x + 6$. An example with three variables is $x^3 + 2xyz^2 - y^2 z + 1$.

1.1.5 Polynomial Remainder Theorem:

In algebra, the polynomial remainder theorem is an application of Euclidean division of polynomials. It states that the remainder of the division of a polynomial f(x) by a linear polynomial (x-r) is equal to f(r). In particular (x-r) is a divisor of f(x) if and only if f(r) = 0. This property is also known as the polynomial factor theorem.

Consider the polynomial $P(x) = 2x^3 - 3x^2 + 4x - 5$ and the value $r = 2$.

Step 1: Divide $P(x)$ by $(x - r)$.

$x - 2\ )\ 2x^3 - 3x^2 + 4x - 5\ (\ 2x^2 + x + 6$

$2x^3 - 4x^2$

$x^2 + 4x - 5$

$x^2 - 2x$

$6x - 5$

$6x - 12$

7

Step 2: The remainder obtained is 7.

According to the Polynomial Remainder Theorem, evaluating $P(r)$ should give the same result.

$P(r) = P(2) = 2 * 2^3 - 3 * 2^2 + 4 * 2 - 5$
$= 16 - 12 + 8 - 5$
$= 7$

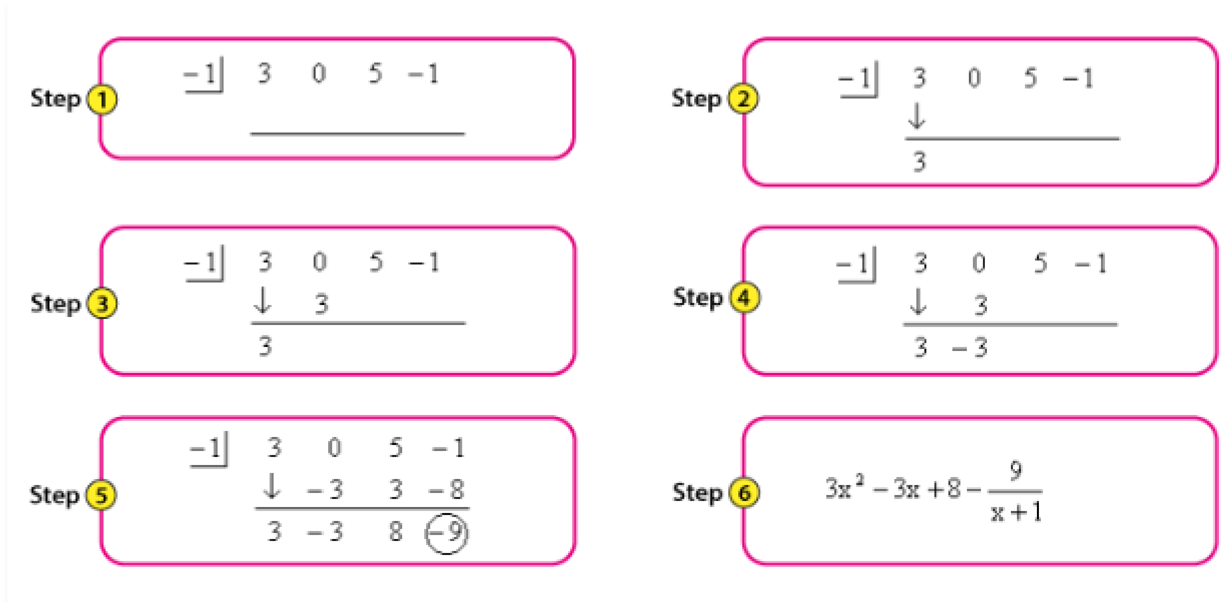
The remainder obtained by dividing $P(x)$ by $(x - 2)$ is 7, and evaluating $P(2)$ also gives

1.1.7 Synthetic division:

In algebra, synthetic division is a method for manually performing Euclidean division of polynomials, with less writing and fewer calculations than long division. The advantages of synthetic division are that it allows one to calculate without writing variables, it uses few

calculations, and it takes significantly less space on paper than long division. Also, subtractions in long division are converted to additions by switching the signs at the very beginning, preventing sign errors. *Fig 1* shows the process.

$Example$: $(2x3+ 5x-1)/(x+1)$



$$\begin{array}{r|rrrr} -1 & 3 & 0 & 5 & -1 \\ & \downarrow & -3 & 3 & -8 \\ \hline & 3 & -3 & 8 & \boxed{-9} \end{array}$$

$$3x^2 - 3x + 8 - \frac{9}{x+1}$$

Fig 1 : Synthetic division

1.2 Challenges

To implement this project I had to face some unique challenges. I used the remainder theorem to simulate how a human would solve the equation using the remainder theorem. Hence there were a lot of unexpected errors and corner cases where I was often stuck.Other challenges :

- To solve the polynomial equation, I used here the Bairstow algorithm for finding real and rational roots.There I faceed challenge for removing epsilon error from roots.
- It was challenge to calculate the complex root of polynomial equation.
- Processing linear equations and generating the necessary intermediate steps to get to the solution.
- Calculating points for plotting the graph of a given linear and polynomial expression.

- It was also a challenge to handle these huge lines of code and different files.

2. Project Overview & Description

2.1 Separate number and variable

Users can input equations in any format. So, we should tokenize this number and variable and the power of the variable. Then store it in a matrix array /vector for calculation and find the solution of linear equation and polynomial equation.

2.2 Solution of linear equation

After tokenizing numbers and variables, we make a matrix and finds determine this matrix. Then replace one column of the matrix with its constant vector and also find determinant of this matrix. We found determinant value for each variable. Here Cramer's rule is used to find the unique solution to these equations.

2.3 Infinite/ No solution exists

If the determinant value of the first matrix is equal to zero then there are infinitely many solutions or no solution exists. Now we check the constant vector, if the constant vector is a multiplicand of the first matrix then there are infinitely many solutions exist. Otherwise, there are no solutions exists.

2.4 Polynomial equation

We use Bairstow method for solving a polynomial equation. To solve the polynomial equation we use a quadratic equation. We divide the main equation with the quadratic equation, if the remainder is equal to zero then we find two roots from the quadratic equation and the main equation will be lower two degrees. We again use the new equation as the main equation and find a new quadratic equation which is also the root of the main equation. We proceed with this until our equation becomes quadratic or with less power. *Fig 2* shows the process :

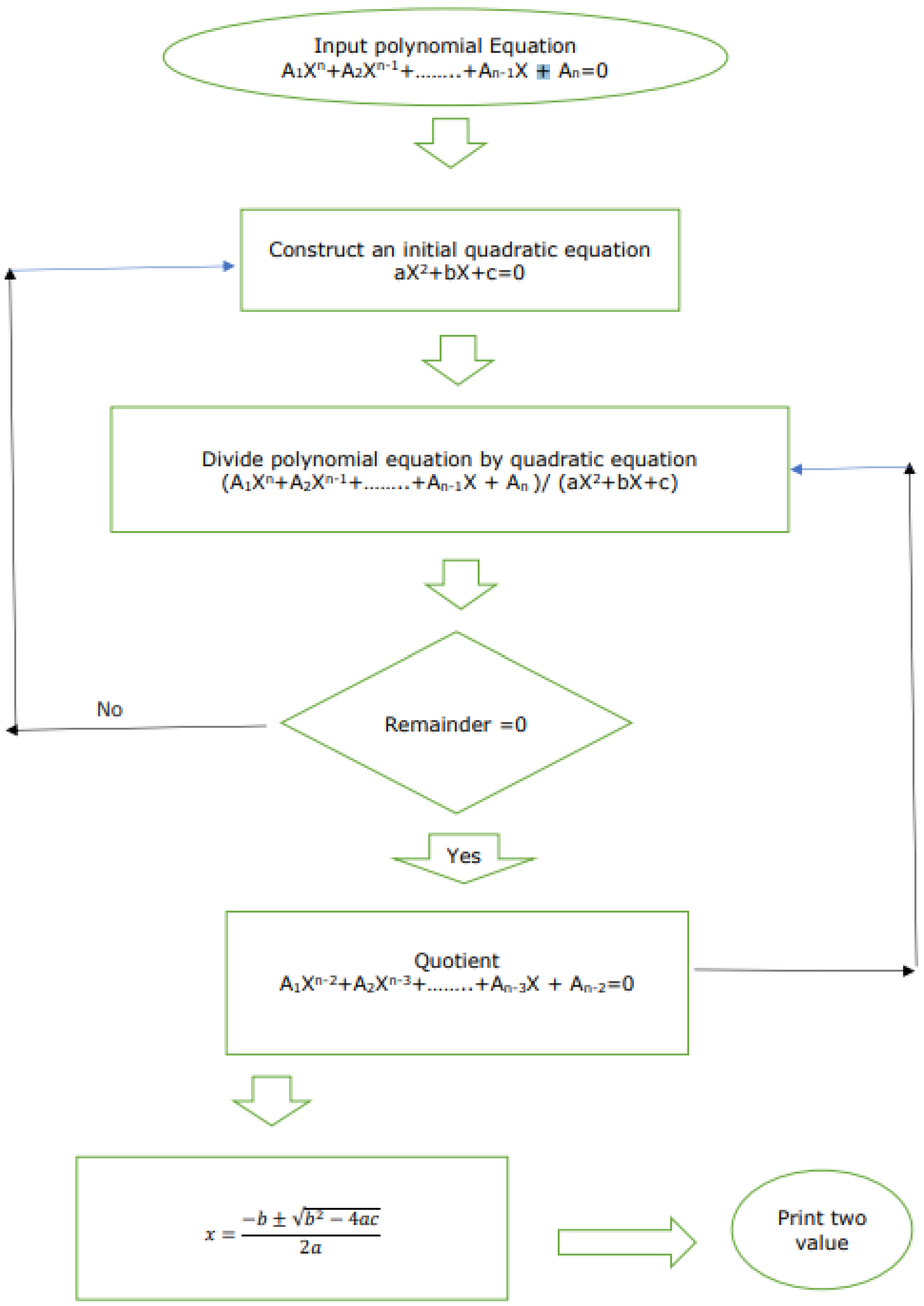

Input polynomial Equation
$A_1X^n+A_2X^{n-1}+........+A_{n-1}X + A_n=0$
Construct an initial quadratic equation
$aX^2+bX+c=0$
Divide polynomial equation by quadratic equation
$(A_1X^n+A_2X^{n-1}+........+A_{n-1}X + A_n)/(aX^2+bX+c)$
Remainder =0
No
Yes
Quotient
$A_1X^{n-2}+A_2X^{n-3}+........+A_{n-3}X + A_{n-2}=0$
$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
Print two value

Fig 2 : Polynomial equation solving flow chart

The program tries to show the steps taken by a human who is trying to analyze a given polynomial expression. Thus while trying to solve a given expression the program frequently outputs the current state of the given expression so the user can keep track of the whole process.

First it takes the user input. After getting the standard string input I had to take the input string and parse the string to separate variables from coefficients and operators, braces and other symbols. Since it manipulates the polynomial expression as simple term, It creates a custom container class named “Term” which contains all of the information of a polynomial term. This makes accessing and manipulating the given expression very simpler and more close to how a human would analyze the expression.

After parsing the terms from given input it determines the degree of the polynomial expression. If polynomial expression has a degree equal or lower than 2 then the sequence of terms are forwarded to “Process Linear Equation” or “Process Quadratic Equation” according to the degree of the given input.(*Fig 3* shows the process).

For a linear equation the program calculates the value of the root. First the constant terms and the terms containing variables are separated. The variables are contained to the left and the constant terms are contained at the right side of the equal sign of the equation. Then the program iteratively adds the variables and constant terms with each other to shorten each side. Then just a simple division at both sides by the variable coefficient gives us the desired root.

For a quadratic equation the program first substitutes all terms to the left side of the equation. Then the program converts the given expression into the standard form of polynomial expressions.
Then the program applies the remainder theorem to find a root for the given expression. After it finds a suitable root for the equation the program converts the equation to a form where it can extract the factor from that equation. Depending on the degree of the remaining polynomial factor the same process may be used as many times as needed. After all extracted factors are of degree 1 the program will return the solution and all the intermediate steps.

Below are the actions taken by the program shown with a simple diagram.(*Fig 3*)

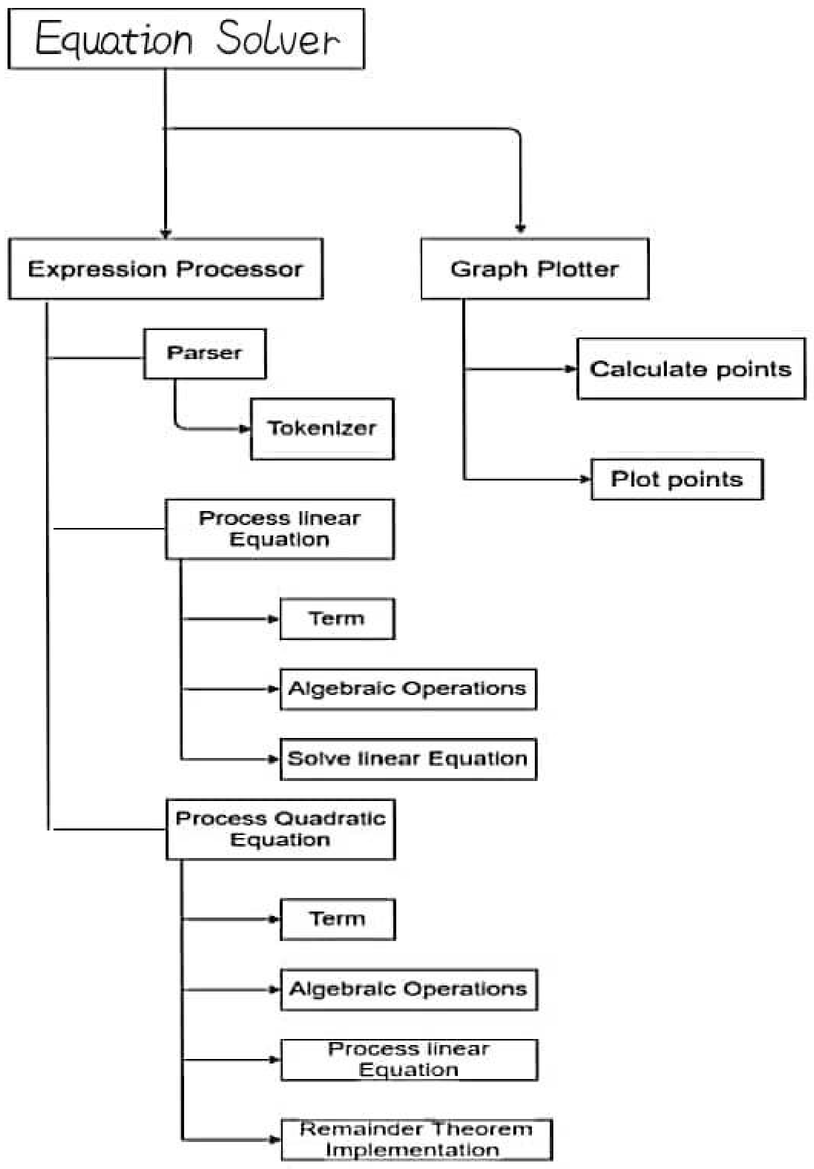
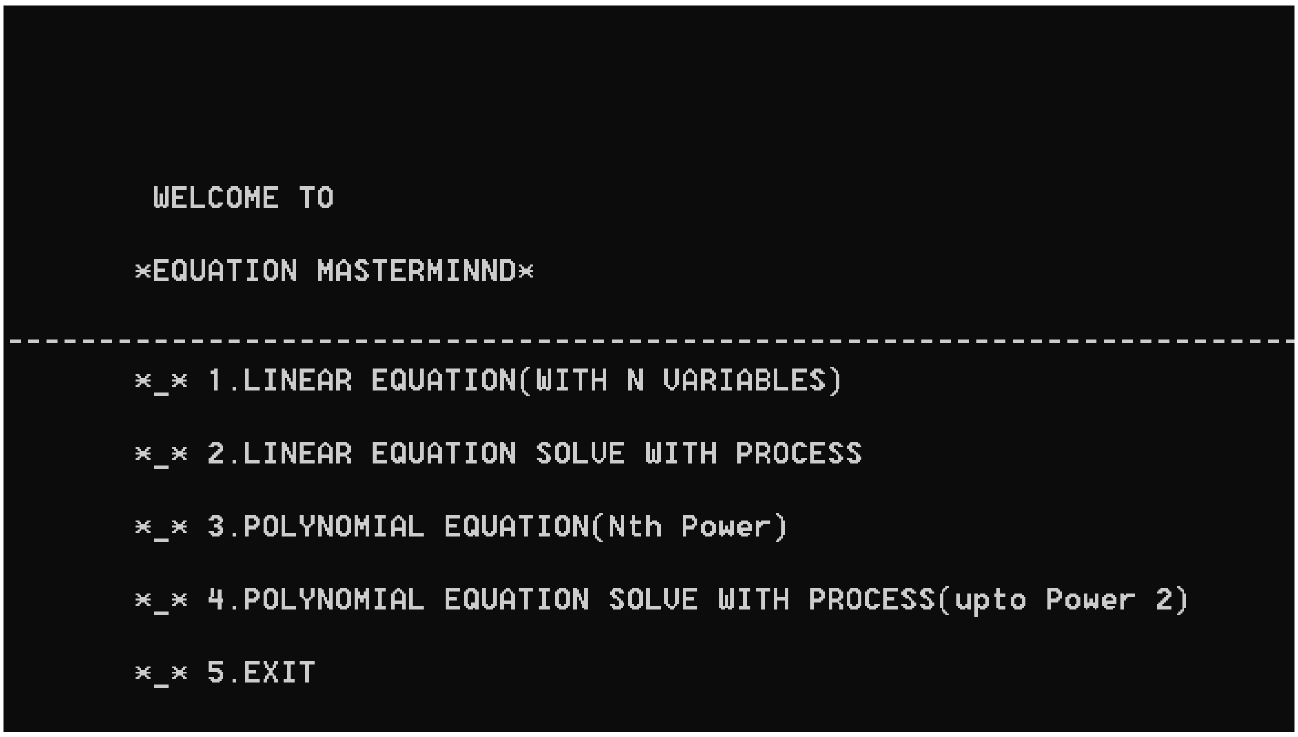


Fig 3: Simple chart showing all the actions taken by the program

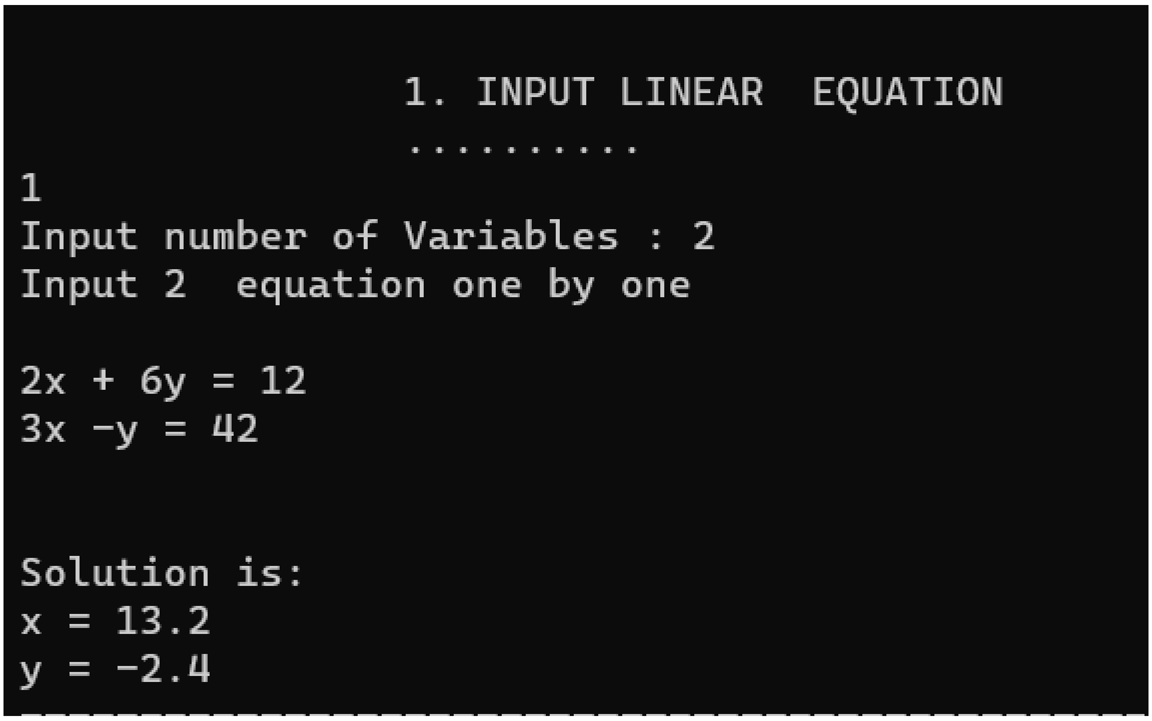
3. User manual

Here users can easily access any feature using the command line.



```
        WELCOME TO

       *EQUATION MASTERMINND*
--------------------------------------------------------------------------
       *_* 1.LINEAR EQUATION(WITH N VARIABLES)

       *_* 2.LINEAR EQUATION SOLVE WITH PROCESS

       *_* 3.POLYNOMIAL EQUATION(Nth Power)

       *_* 4.POLYNOMIAL EQUATION SOLVE WITH PROCESS(upto Power 2)

       *_* 5.EXIT
```

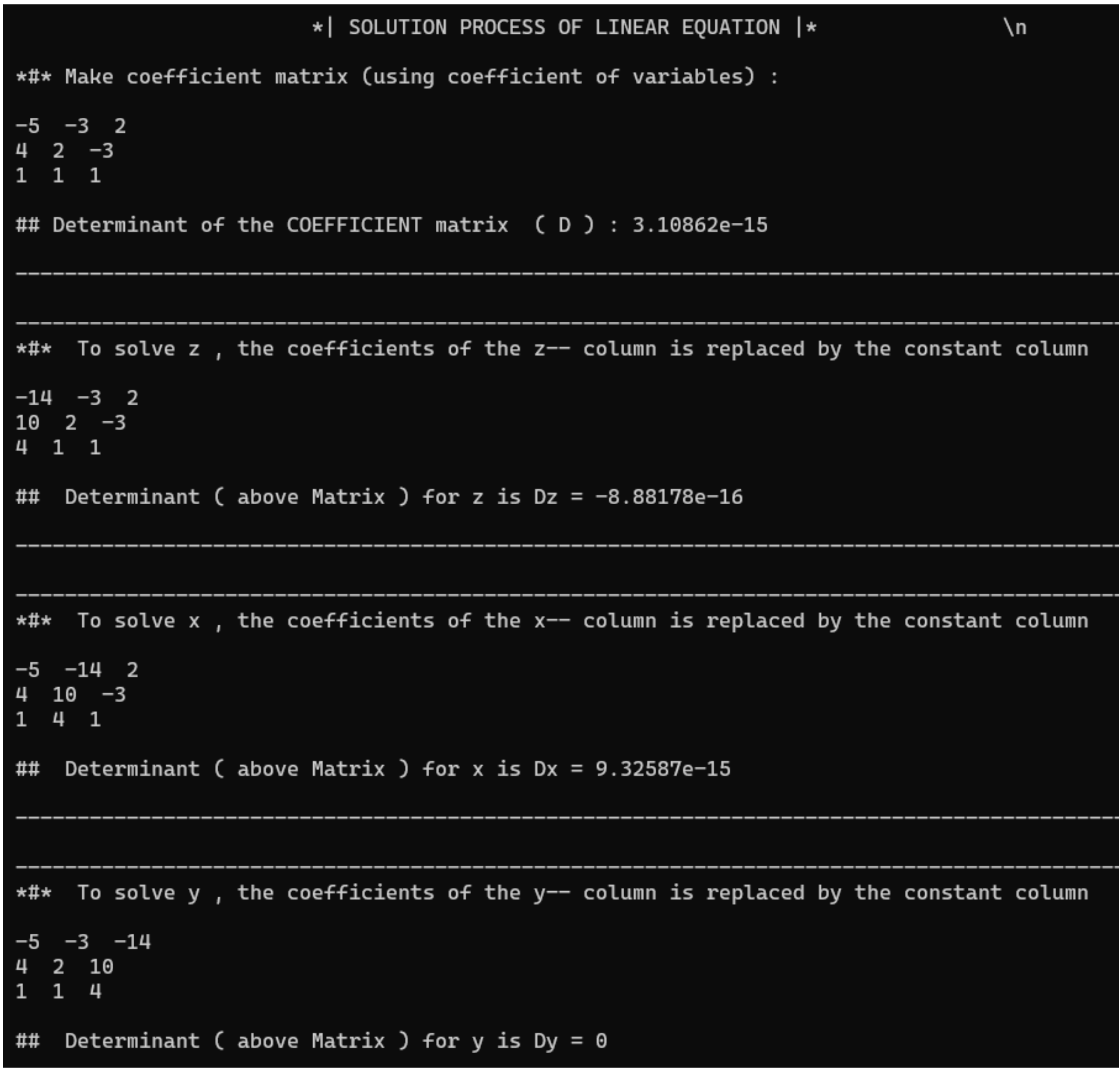
In linear programming, the user can input equations without any modification. then it will take input " The number of variable ". We need same number of equations. then solve it & shows output the roots .



```
                 1. INPUT LINEAR  EQUATION
                 ..........
1
Input number of Variables : 2
Input 2  equation one by one

2x + 6y = 12
3x -y = 42


Solution is:
x = 13.2
y = -2.4
```

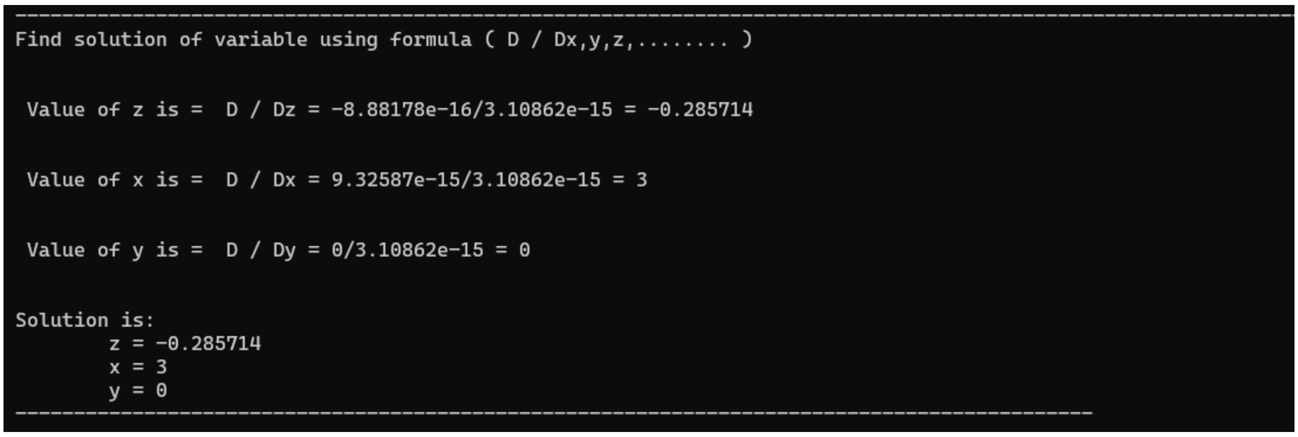
Fig 5: Linear Equation solve

User can see how to solve linear equation step by step using cremer's rule



```
                         *| SOLUTION PROCESS OF LINEAR EQUATION |*              \n

*#* Make coefficient matrix (using coefficient of variables) :

-5  -3  2
4  2  -3
1  1  1

## Determinant of the COEFFICIENT matrix  ( D ) : 3.10862e-15

------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------
*#*  To solve z , the coefficients of the z-- column is replaced by the constant column

-14  -3  2
10  2  -3
4  1  1

##  Determinant ( above Matrix ) for z is Dz = -8.88178e-16

------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------
*#*  To solve x , the coefficients of the x-- column is replaced by the constant column

-5  -14  2
4  10  -3
1  4  1

##  Determinant ( above Matrix ) for x is Dx = 9.32587e-15

------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------
*#*  To solve y , the coefficients of the y-- column is replaced by the constant column

-5  -3  -14
4  2  10
1  1  4

##  Determinant ( above Matrix ) for y is Dy = 0
```

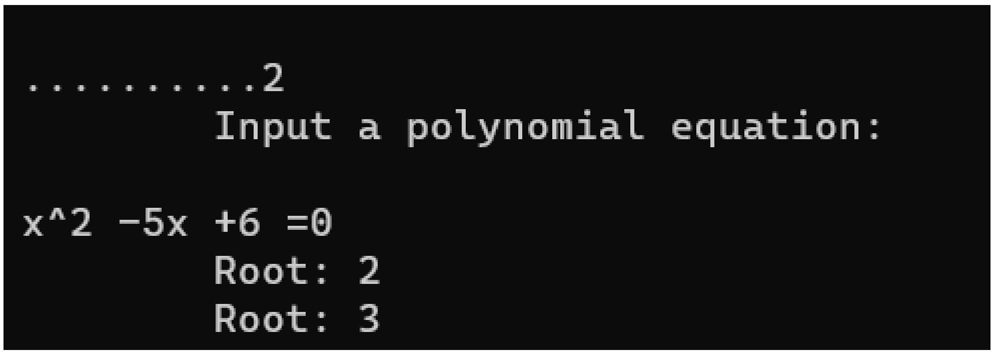
Fig 6: Linear equation solve step by step



```
------------------------------------------------------------------------------------------
Find solution of variable using formula ( D / Dx,y,z,........ )


 Value of z is =  D / Dz = -8.88178e-16/3.10862e-15 = -0.285714


 Value of x is =  D / Dx = 9.32587e-15/3.10862e-15 = 3


 Value of y is =  D / Dy = 0/3.10862e-15 = 0


Solution is:
        z = -0.285714
        x = 3
        y = 0
------------------------------------------------------------------------------------------
```

Fig 7: Linear equation solve step by step

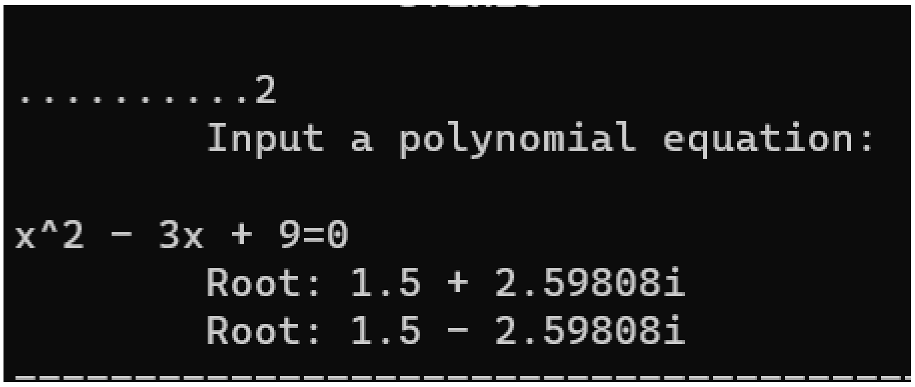
If we select 2 , then it will require “ Polynomial equation” as input. We can input in polynomial expression.



```
..........2
        Input a polynomial equation:

x^2 -5x +6 =0
        Root: 2
        Root: 3
```

Fig 8: Polynomial Equation solution

then click on " Enter " button & get solutions. If there is any complex solution , it will also handle this. See this pic to understand what happens.



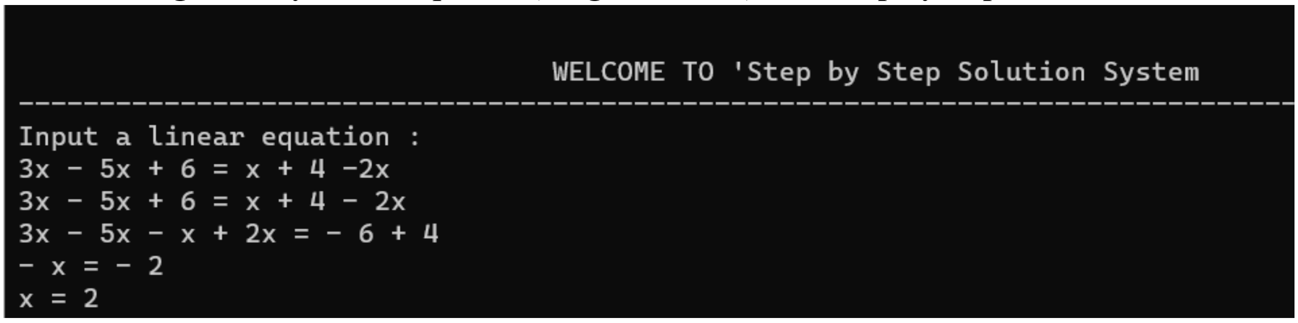
```
..........2
        Input a polynomial equation:

x^2 - 3x + 9=0
        Root: 1.5 + 2.59808i
        Root: 1.5 - 2.59808i
-------------------------------------
```

Fig 9: Polynomial Equation solution(complex root)

After this we will enter " Step By Step Solution System ". Here we will get full solution process of linear equation and quadratic equation solution.

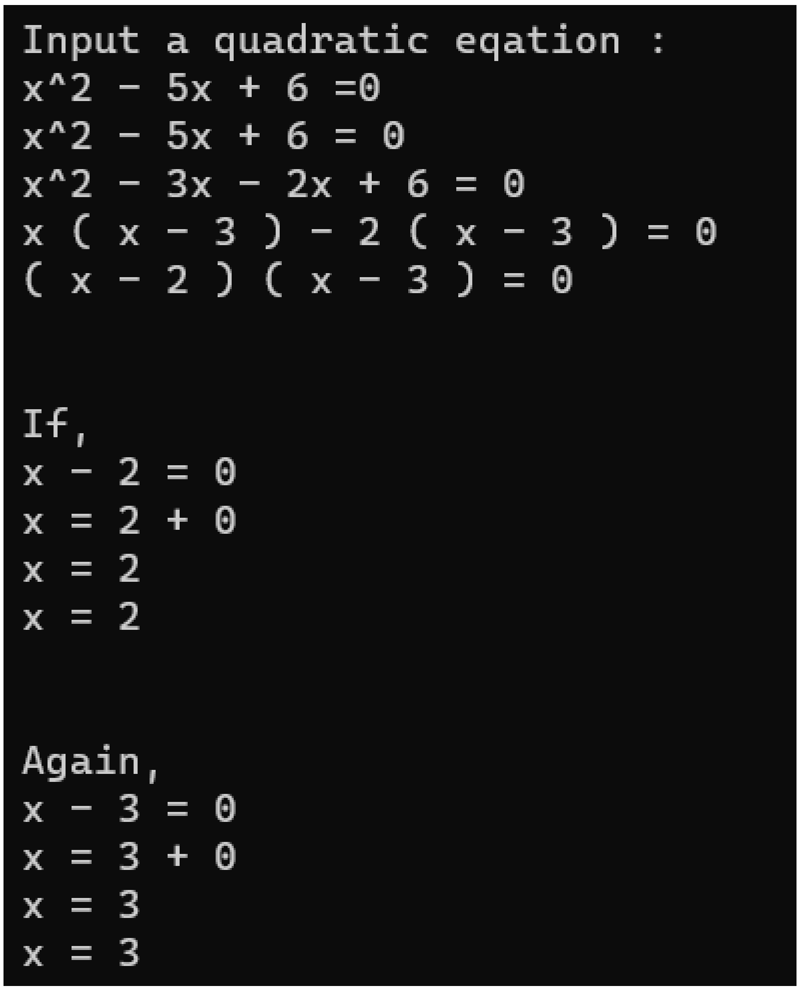
If we input a linear equation , it shows process of solution.

Fig 10: Polynomial equation(single variable) solve step by step



```
                                  WELCOME TO 'Step by Step Solution System
--------------------------------------------------------------------------------
Input a linear equation :
3x - 5x + 6 = x + 4 -2x
3x - 5x + 6 = x + 4 - 2x
3x - 5x - x + 2x = - 6 + 4
- x = - 2
x = 2
```

If user input a quadratic equation , it shows solution process.



```
Input a quadratic eqation :
x^2 - 5x + 6 =0
x^2 - 5x + 6 = 0
x^2 - 3x - 2x + 6 = 0
x ( x - 3 ) - 2 ( x - 3 ) = 0
( x - 2 ) ( x - 3 ) = 0


If,
x - 2 = 0
x = 2 + 0
x = 2
x = 2


Again,
x - 3 = 0
x = 3 + 0
x = 3
x = 3
```

Fig 11: Quadratic equation solve step by step

Besides , we can plot graph of linear function and polynomial fun

4. Conclusion

There are many equations in mathematics, and my project try to solve some selective equations. I try my best to find solutions properly of these equations. I use different types of equations to test this solution. All tests have become successful and it makes a strong proof that this project worked successfully.
From the beginning of this project, I wanted to simulate the process of analyzing a polynomial function using the remainder theorem. The process of finding the factors of a higher degree expression and reducing the function bit by bit manually seen a bit tedious. So in my project I wanted to automate this process. Although limited, I think I have succeeded in simulating the remainder theorem process. Throughout the project I had to face a lot of problems specially at parsing the given input, as anyone can give any combination of terms to generate a polynomial expression .
I tried to learn and implement everything from scratch. Working on this project has helped me have a better idea about windows and network programming. Being new to this programming paradigm, this SPL has challenged my limits. I enjoyed learning on the go and implementing it. I am happy that I could complete the project successfully and have taken on the challenges. I find myself more confident while solving yet the unseen problem, thinking critically, and I believe the experience from this project made me more capable for any future projects that I might work on.

References

 My Git hub repository :
https://github.com/Jihad011/SPL-1

 Polynomial Remainder theorem : https://en.wikipedia.org/wiki/Polynomial_remainder_theorem

Synthetic division : https://en.wikipedia.org/wiki/Synthetic_division

 Bairstow method:
https://math.iitm.ac.in/public_html/sryedida/caimna/transcendental/polynomial%20methods/brs%20method.html

 Polynomial equation : https://www.calculatorsoup.com/calculators/math/math-equation-solver.php

 Linear programming: https://en.wikipedia.org/wiki/Simplex_algorithm

 Cremer’s Rule : 9.8: Solving Systems with Cramer's Rule - Mathematics LibreTexts 14