**EQUATION SOLVER**



**BILAL AHMED 21K-4864**

**ALI ZAIN 21K-4653**

**Computer Organization and Assembly Language**

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES – FAST**

**Introduction:**

Our program allows the user to input either a single polynomial equation of 2nd degree or a series of them for the user to solve them or use them as they see best. For a single equation, the user can either have it’s turning point, outputs for various ‘x’ inputs, or have its roots calculated, whether it be real or imaginary. For a series of equations, the user can have product as well as sum of the equations

**Methodology:**

The software is programmed using Assembly Language which runs using Microsoft Macro Assembler (MASM). The dependencies used are Irvine32 library for its built-in procedures and Macros. Visual Studio Community 2019 and later are utilized for working with the dependencies

Similar logics of high level languages were implemented in x86 language:

Turning Point: The equations are twice derivates in terms of ‘x’ to calculate the co-ordinates of turning point.

Calculating Y: The value of x is first inputted by the user and then substituted in the equation to determine the value of Y.

Determining Roots: The roots, whether they be real or imaginary, will be determined by equating the equation to zero.

Addition of Equations: The equations are simultaneously added by storing the co-efficient of different variables in registers and adding them to form a single summated equation.

Multiplication of Equations: The equations are simultaneously multiplied by storing the co-efficient of different variables in registers and multiplying them to form a single equation which is a product of the series of equations.

**Detailed Design and architecture:**

A menu driver code was written to provide a user friendly interface for the user.

The program helps the user to solve the equations either polynomial or simultaneous. It also provides further operations in both equations.

**Implementation, Testing and Programming code:**

INCLUDE Irvine32.inc

GetSqrRt PROTO, n:DWORD

.data

minus BYTE " -",0

slashh BYTE "/",0

h BYTE 9,9,9," SOLVING EQUATIONS",0Ah,0Ah,0

printxy BYTE 9,9,9,"1. ADDITION OF VARIABLE BY VARIABLE VALUE: ",0

printxy2 BYTE 9,9,9,"2. MULTIPLICATION OF VARIABLE BY VARIABLE VALUE:",0

ent0 byte 9,9,9,"0. RESTART PROGRAM",0

exeet byte 9,9,9,"3. EXIT THIS PROGRAM",0

choice BYTE 9,9,9,"1. POLYNOMIAL EQUATION",0

choice2 BYTE 9,9,9,"2. SIMULTANEOUS EQUATION",0

choice1 BYTE 9,9,9,"ENTER YOUR CHOICE: ",0

enterpoly BYTE 9,9,9,"ENTER A POLYNOMIAL EQUATION: ",0

entersim BYTE 9,9,9,"ENTER TWO LINEAR EQUATIONS: ",0

polyformat BYTE 9,9,9,"EQUATION FORMAT ax^2 + bx + c = Y",0

simulformat BYTE 9,9,9,"EQUATION FORMAT ax + by = c",0

invalid BYTE 9,9,9,"INVALID, ENTER AGAIN: ",0

turn byte 9,9,9,"TURNING POINT X coordinate: ",0

equation1 byte " EQUATION NO 1: ",0

equation2 byte " EQUATUON NO 2: ",0

xsquare BYTE "x^2 ",0

x BYTE "x ",0

y BYTE "y = ",0

equal BYTE " = Y ",0

entera BYTE 9,9,9,"Enter a: ",0

enterb BYTE 9,9,9,"Enter b: ",0

enterc BYTE 9,9,9,"Enter c: ",0

TURNINGPOINT byte 9,9,9,"1. FIND TURNING POINT",0

FINDFACTORS byte 9,9,9,"2. FIND VALUE OF Y FOR CERTAIN X VALUE",0

FINDINGROOTS BYTE 9,9,9,"3. GET ROOTS OF THE EQUATION (ROOTS WOULD BE ROUNDED OFF TO INTEGERS) ",0

exitmenu BYTE 9,9,9,"4. EXIT THE PROGRAM",0

enterx byte 9,9,9,"ENTER VALUE FOR X : ",0

yval BYTE 9,9,9,"VALUE FOR Y= ",0

entersim1 BYTE 9,9,9,"ENTER DATA OF EQ no 1: ",0

entersim2 BYTE 9,9,9,"ENTER DATA OF EQ no 2: ",0

polynomial BYTE " YOUR POLYNOMIAL: ",0

polyA sdword ?

polyB sdword ?

polyC sdword ?

simA1 sdword ?

simB1 sdword ?

simC1 sdword ?

simA2 sdword ?

simB2 sdword ?

simC2 sdword ?

exitt BYTE 9,9,9,"3. EXIT THIS PROGRAM",0

discr SDWORD ?

answerroot BYTE 9,9,9, " ROOTS ARE : ",0

x1 BYTE 0Ah, 9,9,9, "X1 = ",0

x2 BYTE 0Ah, 9,9,9, "X2 = ",0

enterr BYTE 0Ah, 9,9,9,"PRESS ENTER TO PROCEED.....",0

space BYTE " ",0

addresult BYTE 9,9,9,"SUM OF EQUATIONS : " ,0

mulresult BYTE 9,9,9,"MULTIPLICATION OF EQUATIONS COEFFICIENTS : " ,0

.code

main PROC

call start

exit

main endp

start PROC

call clrscr

mov edx , offset h

call writestring

mov edx, offset choice

call writestring

call crlf

mov edx, offset choice2

call writestring

call crlf

mov edx, offset exitt

call writestring

call crlf

mov edx, offset choice1

call writestring

L3:

call readint

cmp eax,3

jg L2

cmp eax, 3

je exittt

cmp eax,1

jl L2

jmp furthercode

L2:

mov edx, offset invalid

call writestring

jmp L3

exittt:

exit

furthercode:

cmp eax, 1

je enterpolyy

cmp eax,2

je enterrsim

enterrsim:

call ENTERSIMULTANEOUSeq

call clrscr

call printSIMUL

call MENUDRIVEN2

enterpolyy:

call ENTERPOLYNOMIAL

call CLRSCR

call PRINTPOLYNOMIAL

call MENUDRIVEN

exit

start endp

ENTERPOLYNOMIAL PROC

call clrscr

mov edx, offset enterpoly

call writestring

call crlf

mov edx, offset polyformat

call writestring

call crlf

mov edx, offset entera

call writestring

call readint

mov polyA, eax

call crlf

mov edx, offset enterb

call writestring

call readint

mov polyB, eax

call crlf

mov edx, offset enterc

call writestring

call readint

mov polyC, eax

call crlf

ret

ENTERPOLYNOMIAL ENDP

PRINTPOLYNOMIAL PROC

mov edx, offset polynomial

call writestring

mov eax, PolyA

call writeint

mov edx, offset xsquare

call writestring

mov eax, PolyB

call writeint

mov edx, offset x

call writestring

mov eax, PolyC

call writeint

mov edx, offset equal

call writestring

call crlf

call crlf

call crlf

mov eax, 0

ret

PRINTPOLYNOMIAL ENDP

MENUDRIVEN PROC

L10000:

call clrscr

L3:

call PRINTPOLYNOMIAL

mov edx, offset TURNINGPOINT

call writestring

call crlf

mov edx, offset FINDFACTORS

call writestring

call crlf

mov edx, offset FINDINGROOTS

call writestring

call crlf

mov edx, offset exitmenu

call writestring

call crlf

mov edx, offset ent0

call writestring

call crlf

mov edx, offset choice1

call writestring

call readdec

cmp eax,1

je TURNPOINT

cmp eax, 3

je ROOTS

cmp eax, 2

je FINDYval

cmp eax,0

je L100

cmp eax, 0

jl L10000

cmp eax,4

je ex

jg L10000

L100:

call start

exit

ROOTS:

call crlf

call CalcDisc

call GetRoots

call crlf

jmp L3

FINDYval:

call clrscr

mov edx, offset enterx

call writestring

call readint

mov esi, eax

imul eax

imul polyA

mov ebx, eax

mov eax, esi

imul polyB

add ebx, eax

add ebx, polyC

mov eax, ebx

mov edx, offset yval

call writestring

call writeint

call crlf

jmp L3

exit

TURNPOINT:

call clrscr

mov edx, offset turn

call writestring

;mov edx, offset minus

;call writestring

mov eax, polyB

neg eax

call writeint

mov edx, offset slashh

call writestring

mov eax, polyA

sal eax , 1

call writeint

call crlf

call crlf

call crlf

jmp L3

ex:

exit

ret

MENUDRIVEN ENDP

ENTERSIMULTANEOUSeq PROC

call clrscr

mov edx, offset entersim1

call writestring

call crlf

mov edx, offset simulformat

call writestring

call crlf

mov edx,offset entera

call writestring

call readint

mov simA1,eax

call crlf

mov edx,offset enterb

call writestring

call readint

mov simB1,eax

call crlf

mov edx,offset enterc

call writestring

call readint

mov simC1,eax

call clrscr

mov edx, offset entersim2

call writestring

call crlf

mov edx, offset simulformat

call writestring

call crlf

mov edx,offset entera

call writestring

call readint

mov simA2,eax

call crlf

mov edx,offset enterb

call writestring

call readint

mov simB2,eax

call crlf

mov edx,offset enterc

call writestring

call readint

mov simC2,eax

ret

ENTERSIMULTANEOUSeq endp

printSIMUL PROC

mov edx, offset equation1

call writestring

mov eax, simA1

call writeint

mov edx, offset x

call writestring

mov eax, simB1

call writeint

mov edx, offset y

call writestring

mov eax, simC1

call writeint

call crlf

mov edx, offset equation2

call writestring

mov eax, simA2

call writeint

mov edx, offset x

call writestring

mov eax, simB2

call writeint

mov edx, offset y

call writestring

mov eax, simC2

call writeint

call crlf

ret

printSIMUL endp

MENUDRIVEN2 PROC

L11:

call clrscr

call printSIMUL

L10:

call crlf

mov edx, offset printxy

call writestring

call crlf

mov edx, offset printxy2

call writestring

call crlf

mov edx, offset exeet

call writestring

call crlf

mov edx, offset ent0

call writestring

call crlf

mov edx, offset choice1

call writestring

call readdec

cmp eax, 1

je addition

cmp eax,2

je multiplication

cmp eax,0

jl L120

cmp eax,3

jg L120

cmp eax,0

je L100

L120:

mov ebx, offset invalid

call writestring

jmp L11

je L100

L100:

call start

exit

multiplication:

call clrscr

call printSIMUL

mov edx , offset mulresult

call writestring

mov eax, simA1

imul simA2

call writeint

mov edx, offset x

call writestring

mov eax, simB1

imul simB2

call writeint

mov edx, offset y

call writestring

mov eax, simC1

imul simC2

call writeint

mov al , 0Ah

call writechar

jmp L10

addition:

call clrscr

call printSIMUL

mov edx , offset addresult

call writestring

mov eax, simA1

add eax, simA2

call writeint

mov edx, offset x

call writestring

mov eax, simB1

add eax, simB2

call writeint

mov edx, offset y

call writestring

mov eax, simC1

add eax, simC2

call writeint

mov al , 0Ah

call writechar

jmp L10

MENUDRIVEN2 endp

CalcDisc PROC

mov eax , polyB

mov ebx , eax

imul ebx

mov discr , eax

mov ebx , 0

mov eax , polyA

mov ebx , 4

mul ebx

mov ebx , polyC

mul ebx

sub discr , eax

ret

CalcDisc ENDP

GetRoots PROC

LOCAL real:SDWORD, image:SDWORD, root1:SDWORD , root2:SDWORD

call PRINTPOLYNOMIAL

cmp discr , 0

jg RealDifferent

je RealEqual

mov eax , polyB

neg eax

cdq

mov ebx , polyA

sal ebx , 1

idiv ebx

mov real , eax

neg discr

INVOKE GetSqrRt, discr

cdq

mov ebx , 0

mov ebx , polyA

sal ebx , 1

idiv ebx

mov image , eax

mov edx , offset answerroot

call writestring

mov edx , offset x1

call writestring

mov eax , real

call writeint

mov edx , offset space

call writestring

mov eax , image

call writeint

mov al , 'i'

call writechar

mov edx , offset x2

call writestring

mov eax , real

call writeint

mov edx , offset space

call writestring

mov eax , image

neg eax

call writeint

mov al , 'i'

call writechar

jmp \_end

RealDifferent:

INVOKE GetSqrRt, discr

mov ebx , polyB

neg ebx

add eax , ebx

cdq

mov ebx , 0

mov ebx , polyA

sal ebx , 1

idiv ebx

mov root1 , eax

mov eax , 0

INVOKE GetSqrRt, discr

mov ebx , polyB

neg ebx

sub ebx , eax

mov eax , ebx

cdq

mov ebx , 0

mov ebx , polyA

sal ebx , 1

idiv ebx

mov root2 , eax

jmp answer

RealEqual:

mov ebx , polyA

sal ebx , 1

mov eax , polyB

neg eax

cdq

idiv ebx

mov root1 , eax

mov root2 , eax

answer:

mov edx , offset answerroot

call writestring

mov edx , offset x1

call writestring

mov eax , root1

call writeint

mov edx , offset x2

call writestring

mov eax , root2

call writeint

\_end:

mov edx , offset enterr

call writestring

call readchar

ret

GetRoots ENDP

GetSqrRt PROC, n:DWORD

LOCAL z:DWORD, i:DWORD, term:DWORD

mov eax , 0

mov eax , n

mov i , 0

mov z , 0

mov term , 0

mov z , eax

shr eax , 1

mov term , eax

L1:

mov ecx , i

cmp ecx , term

jge \_outofloop

mov eax , n

mov edx , 0

mov ebx , z

div ebx

add eax , z

shr eax , 1

mov z , eax

inc i

Jmp L1

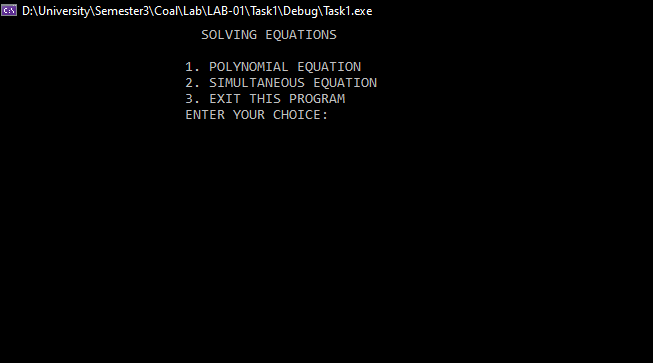
\_outofloop:

ret

GetSqrRt ENDP

end main

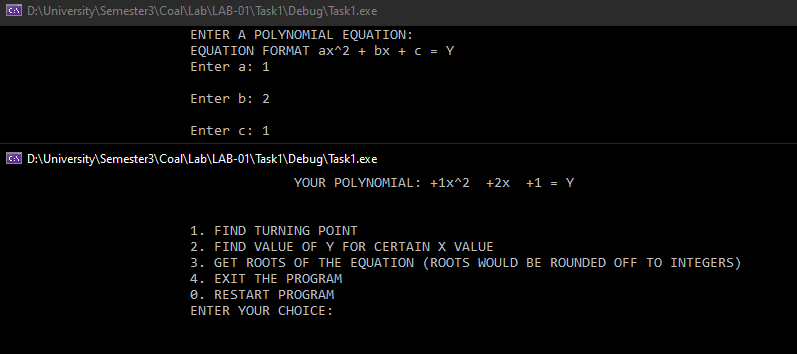
**Results/ Software Simulation and discussion:**



Choose any option from the main menu,

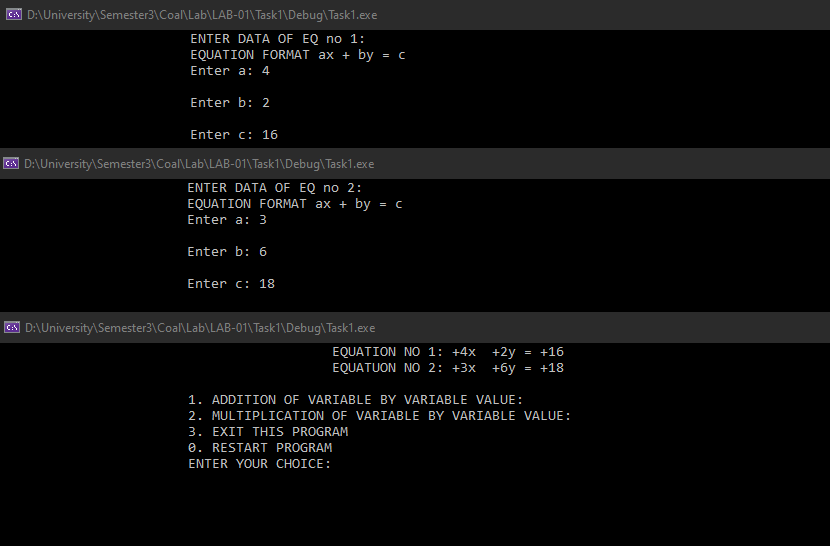
In Polynomial Equation, user must input the values, then the program will give user option to

1. Calculate the turning point of the equation
2. Calculate the value of y for all the value of x
3. Get the roots of the equation



In Simultaneous Equation, user must input the values, then the program will give user option to

1. Addition
2. Multiplication



**Conclusion, Cost and future work:**

Despite the algorithms we used might be common and readily used, implementation of them in x86 language is not very common in the most user-friendly format. We would like to implement more functions in this project and try to replicate the working of an actual scientific calculator. We will try to implement a user-friendly GUI to the project as well as launch our project code in various platforms for different enthusiastic users