**Introduction:**

Calculators are a big help when doing mathematical equations correctly. They are also a useful tool in learning different ways to do mathematics. The use of them plays a big part in excelling in math.

**Methodology:**

We will use the concepts of assembly language that we have learnt in COAL Lab.

We will use emu8086 as a compiler in this project.

**Operations:**

We have make various operations in our Calculator.

1. Addition
2. Multiplication
3. Subtraction
4. Division
5. Cube
6. Square

**Code:**

org 100h

include emu8086.inc

.data

a dw ?

num1 dw ? ;FIRST OPERAND

num2 dw ? ;SECOND OPERAND

result dw ?

.code

printn

print '<<<<+-\*/WELCOME+-\*/>>>>'

printn

printn

printn

print '<<<<CODE BY 200901057 , 200901060>>>>>'

printn

printn

printn

printn

print '<<<<<<What Operation You Want To Perform:>>>>>>> '

printn

printn ;OPERATIONS TO BE PERFORMED

print '1>> Addition (+) '

printn

printn

print '2>> Multiplication(\*) '

printn

printn

print '3>> Subtraction (-) '

printn

printn

print '4>> Division (/) '

printn

printn

print '5>> Square (^2) '

printn

printn

print '6>> Cube (^3) '

printn

printn

print '7>> Exit'

printn

printn

print ' Note:Integers Value only'

start:

printn

printn

print '>> Press Any Number From The Above Given List :'

call scan\_num

mov a,cx

mov ax,a

printn

;COMPARE THE ENTERED NUMBER WITH THE FOLLOWING CONDITIONS

;JUMP TO THE ENTERED CONDITION

cmp ax,1

je addition

cmp ax,2

je multiplication

cmp ax,3

je subtraction

cmp ax,4

je division

cmp ax,5

je square

cmp ax,6

je cube

cmp ax,7

je exit

**ADDITION**

addition:

printn

print '++++++++ADDITION++++++ ' ; PERFORMING ADDITION

printn

printn

print '>> Enter First Number: ' ;FIRST OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

print '>> Enter Second Number: ' ;SECOND OPERAND

call scan\_num

mov num2,cx

mov bx,num2

printn

add ax,bx

print '>> Result: ' ;RESULT

call print\_num

jmp start:

**MULTIPLICATION**

multiplication:

printn

print '\*\*\*\*\*\*MULTIPLICATION\*\*\*\*\*\*\* ' ; PERFORMING MULTIPLICATION

printn

printn

print '>> Enter First Number: ' ;FIRST OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

print '>> Enter Second Number: ' ;SECOND OPERAND

call scan\_num

mov num2,cx

mov bx,num2

printn

mul bx ;RESULT

print '>> Result: '

call print\_num

jmp start:

**SUBTRACTION**

subtraction:

printn

print '----------SUBTRACTION--------- ' ;PERFORMING SUBTRACTION

printn

printn

print '>> Enter First Number: ' ;FIRST OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

print '>> Enter Second Number: ' ;SECOND OPERAND

call scan\_num

mov num2,cx

mov bx,num2

printn

sub ax,bx

print '>> Result: ' ;RESULT

call print\_num

jmp start:

**DIVISION**

division:

printn

print '////////DIVISION///////////' ;PERFORMING DIVISION

printn

print '>> Enter First Number: ' ;FIRST OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

print '>> Enter Second Number: ' ;SECOND OPERAND

call scan\_num

mov num2,cx

mov bx,num2

printn

div bx

PRINT '>> Result: ' ;RESULT

call print\_num

jmp start:

**SQUARE**

square:

printn

print '^2^2^2^2 SQUARE ^2^2^2^2 ' ;PERFORMING SQUARE

printn

printn

print '>> Enter The Number: ' ;OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

mul ax

print '>> Result: '

call print\_num ;RESULT

jmp start:

**CUBE**

cube:

printn

print '^3^3^3^3 CUBE ^3^3^3^3' ;PERFORMING CUBE

printn

printn

print '>> Enter The Number: ' ;OPERAND

call scan\_num

mov num1,cx

mov ax,num1

printn

mul num1

mov result,ax

mul num1

print '>> Result: ' ;RESULT

call print\_num

jmp start:

**EXIT**

exit:

;EXIT

printn

print '<<<<<<+-\*/! GOOD BYE !+-\*/>>>>> '

printn

DEFINE\_SCAN\_NUM

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

Ret

**OUTPUT**

**Main Menu**

Text

Description automatically generated

**ADDITION**

Text

Description automatically generated

**SUBTRACTION**

Text

Description automatically generated

**MULTIPLICATION**

Text

Description automatically generated

**DIVISION**

Text

Description automatically generated

**SQUARE**

Text

Description automatically generated

**CUBE**

Text

Description automatically generated

**GOODBYE**

