*CALCULATOR USING*

*ASSEMBLY LANGUAGE*

* INTRODUCTION:

Assembly language is a low level language containing symbols and it is a human readable language. Assembly language can directly communicate with the computer’s hardware. On the other hand, calculators are used to solve any mathematical problems.

As assembly language has features like labels, comments, operand, registers and operations (assembler directives and mnemonics). By using these features A Simple Calculator has been designed.

* OVER VIEW ABOUT THE PROJECT:

This calculator can perform functions like Addition, Subtraction, Multiplication, Division, Percentage and power. Two new modes Increment and Decrement have been introduced in the Calculator. Which are rarely found in any calculator.

Different mnemonics and commands especially the use of loops and jump statements were beneficial in creating functions in order to run the code in a simple and efficient way.

* SCOPE:

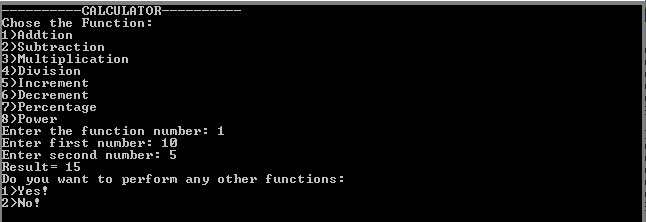
***what was the purpose of creating this calculator?***

The purpose for designing it was for the users to solve any type of problems. Rather than doing long calculations a user can simply use a calculator with unique methods to solve his problem within seconds. It is time efficient All this have been made easy because of Assembly language because,

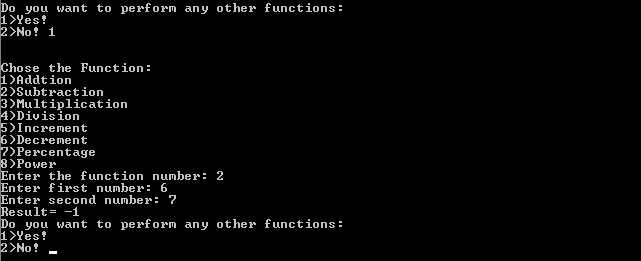
* It allows complex problems to run in simpler ways
* It is memory efficient.
* Requires less instructions to get the results.
* CODE:

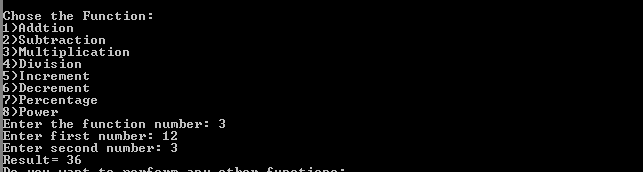
|  |
| --- |
| org 100h  include "emu8086.inc"  .model small  .stack 100h  .data  msg0 db "----------CALCULATOR----------$"  msg1 db "Enter first number: $"  msg2 db "Enter second number: $"  msg3 db "Result =$ "  msg4 db "Chose the Function:$"  msg5 db "Wrong operator! $"  msg6 db "1)Addtion $"  msg7 db "2)Subtraction $"  msg8 db "3)Multiplication $"  msg9 db "4)Division $"  msg10 db "5)Increment $"  msg11 db "6)Decrement $"  msg12 db "Enter the function number: $"  msg13 db "Result= $"  msg14 db "Do you want to perform any other functions: $"  msg15 db "1>Yes! $"  msg16 db "2>No! $"  msg17 db "Enter number: $"  msg18 db "Enter total number: $"  msg19 db "Enter number obtained: $"  msg20 db "7)Percentage $"  msg21 db "THANK YOU! $"  msg22 db "8)Power $"  msg23 db "Enter the power: $"  operator dw ?  num1 dw ?  num2 dw ?  result dw ?  opt dw ?  endcondition dw 1d  .code  main proc    MOV ax,@data  MOV ds,ax  LEA dx,msg0  MOV ah,09h  INT 21h    while:  printn      wrongfunction:  ;CHOSE THE FUNCTION  LEA dx,msg4  MOV ah,09h  INT 21h  printn  ;ADDTION MSG  LEA dx,msg6  MOV ah,09h  INT 21h  printn  ;SUBTRACTION MSG  LEA dx,msg7  MOV ah,09h  INT 21h  printn  ;MULTIPLICATION MSG  LEA dx,msg8  MOV ah,09h  INT 21h  printn  ;DIVISION MSG  LEA dx,msg9  MOV ah,09h  INT 21h  printn  ;INCREMENT MSG  LEA dx,msg10  MOV ah,09h  INT 21h  printn  ;DECREMENT MSG  LEA dx,msg11  MOV ah,09h  INT 21h  printn  ;PERCENTAGE MSG  LEA dx,msg20  MOV ah,09h  INT 21h  printn    ;POWER MSG  LEA dx,msg22  MOV ah,09h  INT 21h  printn      LEA dx,msg12  MOV ah,09h  INT 21h        CALL scan\_num  MOV operator,cx    CMP operator,1  JE addition    CMP operator,2  JE subtraction    CMP operator,3  JE multiplication    CMP operator,4  JE division    CMP operator,5  JE increment    CMP operator,6  JE Decrement    CMP operator,7  JE percentage    CMP operator,8  JE power  ;wrong operator jumping to start of the function  printn  LEA dx,msg5  MOV ah,09h  INT 21h  printn  JMP wrongfunction      ;ADDITION  addition:  printn  ;TAKING FIRST NUMBER FROM THE USER  LEA dx,msg1  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  printn  ;TAKING SECOND NUMBER FROM THE USER  LEA dx,msg2  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  MOV ax,num1  ADD ax,num2  MOV result,ax  JMP resl      ;SUBTRACTION  subtraction:  printn  ;TAKING FIRST NUMBER FROM THE USER  LEA dx,msg1  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  printn  ;TAKING SECOND NUMBER FROM THE USER  LEA dx,msg2  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  MOV ax,num1  SUB ax,num2  MOV result,ax  JMP resl    ;MUlTIPLICATION  multiplication:  printn  ;TAKING FIRST NUMBER FROM THE USER  LEA dx,msg1  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  printn  ;TAKING SECOND NUMBER FROM THE USER  LEA dx,msg2  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  MOV ax,num1  MUL num2  MOV result,ax  JMP resl    ;DIVISION FUNCTION  division:  printn  ;TAKING FIRST NUMBER FROM THE USER  LEA dx,msg1  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  printn  ;TAKING SECOND NUMBER FROM THE USER  LEA dx,msg2  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  MOV dx,0  MOV ax,num1  IDIV num2  MOV result,ax  JMP resl      ;INCREMENT  increment:  printn  ;TAKING NUMBER FROM THE USER  LEA dx,msg17  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  ADD num1,1  MOV ax,num1  MOV result,ax  JMP resl      ;DECREMENT  Decrement:  printn  ;TAKING FIRST NUMBER FROM THE USER  LEA dx,msg17  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  SUB num1,1  MOV ax,num1  MOV result,ax  JMP resl    ;PERCENTAGE  percentage:  printn  ;TAKING TOTAL NUMBER FROM THE USER  LEA dx,msg18  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  printn  ;TAKING NUMBER OBTAINED FROM THE USER  LEA dx,msg19  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  MOV dx,0  MOV ax,num1  IDIV num2  MOV bx,100  MUL bx  MOV result,ax  JMP resl      ;POWER  power:  ;TAKING NUMBER FROM THE USER  printn  LEA dx,msg17  MOV ah,09h  INT 21h  CALL scan\_num  MOV num1,cx  printn  LEA dx,msg23  MOV ah,09h  INT 21h  CALL scan\_num  MOV num2,cx  MOV bx,num1  loo:  MOV ax,bx  MUL num1  MOV num1,ax  SUB num2,1  CMP num2,1  JG loo  MOV ax,num1  MOV result,ax  jmp resl      ;DISPLAYING RESULT  resl:  printn  LEA dx,msg13  MOV ah,09h  INT 21h  MOV ax,result  CALL print\_num    ;ASKING TO START THE FUNCTION  printn  LEA dx,msg14  MOV ah,09h  INT 21h  printn  ;YESS  LEA dx,msg15  MOV ah,09h  INT 21h  printn  ;NO  LEA dx,msg16  MOV ah,09h  INT 21h  CALL Scan\_num  MOV opt,cx  printn  printn  CMP opt,1  JE while      ;DISPLAYING TANK YOU AND ENDING  printn  LEA dx,msg21  MOV ah,09h  INT 21h  printn    ENDP  DEFINE\_SCAN\_NUM  DEFINE\_PRINT\_NUM  DEFINE\_PRINT\_NUM\_UNS  hlt  ret |

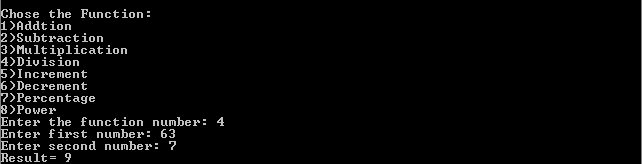
* Test/output result of different functions.
* Addition Function:



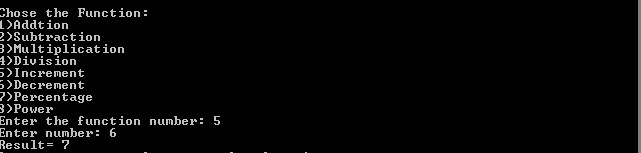
* Subtraction Function;



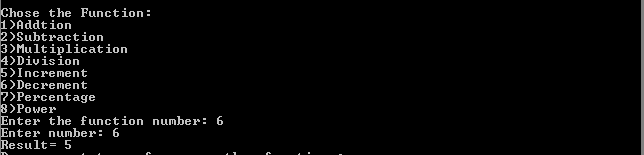
* Multiplication Function:
* Division Function:



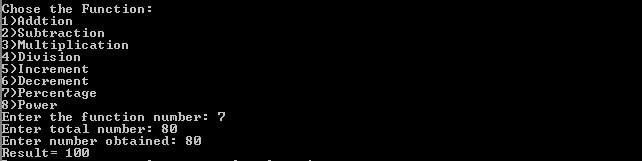
* Increment Function:



* Decrement Function:

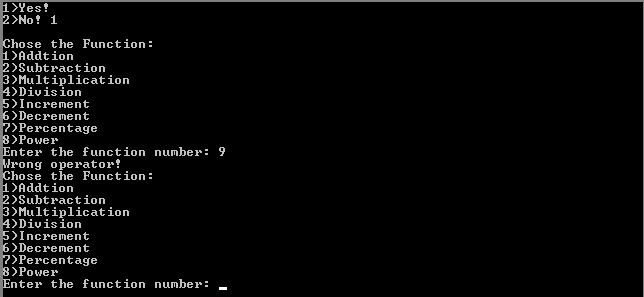


* Percentage Function:



* Power Function;



* In case if the user types a wrong operator value e.g. (9)

***The option of choosing the function will be asked again due to the jump condition***.

* CONCLUSION:

The purpose of designing calculator was to do correct calculation efficiently. It should give user a relieve from doing the mental calculation and to need to rely on paper calculations.

**…………………………………………………………………………………………….**