.MODEL SMALL

.DATA

menu1 DB "Select an operation (1-7): $", 0

operation DB "Operation selected: $", 0

result\_msg DB "Result: $", 0

newline DB 0Dh, 0Ah, 0

temp DW ?

res DW ?

;Messages

THANK DB 0AH, 0AH, 0AH, 0DH, " Thank you for using our application $"

INVALID DB 0AH, 0AH, 0DH, " Invalid input. Please choose a different option.$"

CHOOSE DB 0AH, 0AH, 0DH, " Enter option: $"

SUCCESS DB 0AH, 0AH, 0DH, " RESULT $ "

;Withdraw

num1 DB 0AH, 0AH, 0DH, " Enter number1 $ "

num2 DB 0AH, 0AH, 0DH, " Enter number2 $ "

INVALID\_INPUT DB 0AH, 0AH, 0DH, " INVALID INPUT"

;Digit place

TH DW 10000

T DW 10

.CODE

MAIN PROC

MOV AX, DATA

MOV DS, AX

;Display the menu

MENU:

MOV AH, 09H

LEA DX, menu1

INT 21H

CHOOSE\_OPTION: MOV AH, 09H

LEA DX, CHOOSE

INT 21H

MOV AH, 01H

INT 21H

CMP AL, 49

JE do\_add

CMP AL, 50

JE do\_sub

CMP AL, 51

JE do\_mul

CMP AL, 52

JE do\_div

CMP AL, 53

JE do\_neg

CMP AL, 54

JE do\_inc

CMP AL, 55

JE do\_dec

cmp al,56

JE Exit

jmp INP\_ERROR

do\_add:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

push bx

MOV AH, 09H

LEA DX, num2

INT 21H

CALL INPUT\_4DIGIT\_NUM

pop ax

add ax,bx

mov res,ax

mov temp,1

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_sub:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

push bx

MOV AH, 09H

LEA DX, num2

INT 21H

CALL INPUT\_4DIGIT\_NUM

pop res

cmp res,bx

JL l3

sub res,bx

mov temp,1

JMP l2

l3:

mov temp,0

xchg res,bx

sub res,bx

l2:

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_mul:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

push bx

MOV AH, 09H

LEA DX, num2

INT 21H

CALL INPUT\_4DIGIT\_NUM

pop ax

cmp ax,0

JL l5

cmp bx,0

JL l5

JMP l6

l5:

mov temp,0

l6:

mov temp,1

mul bx

mov res,ax

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_div:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

push bx

MOV AH, 09H

LEA DX, num2

INT 21H

CALL INPUT\_4DIGIT\_NUM

pop ax

cmp ax,0

JL l7

cmp bx,0

JL l7

JMP l8

l7:

mov temp,0

l8:

mov temp,1

div bx

mov res,ax

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_neg:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

;neg bx

mov res,bx

mov temp,0

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_inc:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

inc bx

mov res,bx

mov temp,1

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

do\_dec:

MOV AH, 09H

LEA DX, num1

INT 21H

CALL INPUT\_4DIGIT\_NUM

dec bx

mov res,bx

CALL SUCCESS\_MSG

JMP CHOOSE\_OPTION

;Exit the application

EXIT: MOV AH, 09H

LEA DX, THANK

INT 21H

MOV AH, 4CH

INT 21H

;Invalid input

INP\_ERROR: MOV AH, 09H

LEA DX, INVALID

INT 21H

JMP CHOOSE\_OPTION

;Procedure to input a 4-digit number

;Procedure to input a 4-digit number

;Procedure to input a 4-digit number

INPUT\_4DIGIT\_NUM PROC

XOR BX, BX

MOV CX, 4

MOV TH, 1000

READ\_DIGIT: MOV AH, 01H

INT 21H

CMP AL, 0DH ; Check if the user pressed Enter

JE ENTER\_PRESSED ; Jump if Enter was pressed

SUB AL, '0'

MOV AH, 0

MUL TH

ADD BX, AX

;PUSH AX

MOV AX, TH

DIV T

MOV TH, AX ;th = 100

;POP AX ;AX =

LOOP READ\_DIGIT

ENTER\_PRESSED:

; Code to execute when Enter is pressed

MOV AX, BX

CMP CX,0

JE L1

LABEL1:

DIV T

LOOP LABEL1

L1:

MOV BX, AX

RET

INPUT\_4DIGIT\_NUM ENDP

;Procedure to display a number

PRINT\_NUMBER PROC

; Check if the number is negative

PUSH AX ; Save AX

PUSH BX ; Save BX

PUSH CX ; Save CX

XOR CX, CX ; Clear CX (used for counting digits)

MOV BX, 10 ; Base 10 divisor

CMP temp, 0

JNE PRINT\_POSITIVE

mov temp,ax ; If AX >= 0, skip the negative handling

; Handle negative number

MOV AH, 02H ; Print character function

MOV DL, '-' ; Set DL to '-' for the negative sign

INT 21H ; Print '-'

NEG AX ; Convert AX to positive for printing

mov ax,temp

PRINT\_POSITIVE:

CONVERT\_LOOP:

XOR DX, DX ; Clear DX

DIV BX ; Divide AX by 10, remainder in DX

PUSH DX ; Store remainder (digit) on stack

INC CX ; Increment digit count

CMP AX, 0 ; Check if AX is zero

JNE CONVERT\_LOOP ; Repeat if not

PRINT\_DIGITS:

POP DX ; Retrieve digit from stack

ADD DL, '0' ; Convert to ASCII

MOV AH, 02H ; Print character function

INT 21H ; Print digit

LOOP PRINT\_DIGITS ; Repeat for all digits

POP CX ; Restore CX

POP BX ; Restore BX

POP AX ; Restore AX

RET

PRINT\_NUMBER ENDP

;Procedure to display a success message

SUCCESS\_MSG PROC

MOV AH, 09H

LEA DX, SUCCESS

INT 21H

XOR AX, AX

MOV AX, res

CALL PRINT\_NUMBER

RET

SUCCESS\_MSG ENDP

END MAIN