```
org 100h

jmp start       ; jump over data declarations

; ------------------------
; DATA SEGMENT
; ------------------------
msg:    db "1-Add",0dh,0ah,"2-Multiply",0dh,0ah,"3-Subtract",0dh,0ah,"4-Divide",0Dh,0Ah, '$'
msg2:   db 0dh,0ah,"Enter First No : $"
msg3:   db 0dh,0ah,"Enter Second No : $"
msg4:   db 0dh,0ah,"Choice/Error: Invalid input or divide by zero! $"
msg5:   db 0dh,0ah,"Result : $"
msg6:   db 0dh,0ah,'Thank you for using the calculator! Press any key...', 0Dh,0Ah, '$'

; ------------------------
; CODE START
; ------------------------
start:
    mov ah,9
    mov dx, offset msg
    int 21h

    mov ah,0
    int 16h
    cmp al,31h
    je Addition
    cmp al,32h
    je Multiply
    cmp al,33h
    je Subtract
    cmp al,34h
    je Divide

    ; Invalid input
    mov ah,09h
    mov dx, offset msg4
    int 21h
    mov ah,0
    int 16h
    jmp start

; ------------------------
; ADDITION
; ------------------------
Addition:
    mov ah,09h
    mov dx, offset msg2
    int 21h
    mov cx,0
    call InputNo
    push dx

    mov ah,9
    mov dx, offset msg3
    int 21h
    mov cx,0
    call InputNo
    pop bx
    add dx,bx
    push dx

    mov ah,9
    mov dx, offset msg5
    int 21h

    mov cx,10000
    pop dx
    call View
    jmp exit

; ------------------------
; SUBTRACTION (with sign)
; ------------------------
Subtract:
    mov ah,09h
    mov dx, offset msg2
    int 21h
    mov cx,0
    call InputNo
    push dx

    mov ah,9
    mov dx, offset msg3
    int 21h
    mov cx,0
    call InputNo
    pop bx

    xor si, si
    cmp bx,dx
    jge do_subtract
    xchg bx, dx
    mov si, 1

do_subtract:
    sub bx, dx
    mov dx, bx
    push dx

    mov ah,9
    mov dx, offset msg5
    int 21h

    cmp si, 0
    je print_result
    mov dl, '-'
    mov ah, 2
    int 21h

print_result:
    mov cx,10000
    pop dx
    call View
    jmp exit

; ------------------------
; MULTIPLICATION
; ------------------------
Multiply:
    mov ah,09h
    mov dx, offset msg2
    int 21h
    mov cx,0
    call InputNo
    push dx

    mov ah,9
    mov dx, offset msg3
    int 21h
    mov cx,0
    call InputNo
    pop bx

    mov ax,dx
    mul bx
    mov dx,ax
    push dx

    mov ah,9
    mov dx, offset msg5
    int 21h
    mov cx,10000
    pop dx
    call View
    jmp exit

; -----------------------
; DIVISION (with divide-by-zero handling)
; -----------------------
Divide:
    mov ah,09h
    mov dx, offset msg2
    int 21h
    mov cx,0
    call InputNo
    push dx

    mov ah,9
    mov dx, offset msg3
    int 21h
    mov cx,0
    call InputNo
    cmp dx,0
    je divide_error
    pop bx

    mov ax,bx
    mov cx,dx
    xor dx,dx
    div cx
    mov bx,dx
    mov dx,ax
    push bx
    push dx

    mov ah,9
    mov dx, offset msg5
    int 21h

    mov cx,10000
    pop dx
    call View

    pop bx
    cmp bx,0
    je exit
    jmp exit

divide_error:
    mov ah,9
    mov dx, offset msg4
    int 21h
    mov ah,0
    int 16h
    jmp start

; -----------------------
; INPUT NUMBER ROUTINE
; -----------------------
InputNo:
    mov ah,0
    int 16h
    mov dx,0
    mov bx,1
    cmp al,0dh
    je FormNo
    sub ax,30h
    call ViewNo
    mov ah,0
    push ax
    inc cx
    jmp InputNo

FormNo:
    pop ax
    push dx
    mul bx
    pop dx
    add dx,ax
    mov ax,bx
    mov bx,10
    push dx
    mul bx
    pop dx
    mov bx,ax
    dec cx
    cmp cx,0
    jne FormNo
    ret

; ------------------------
; DISPLAY DIGITS
; ------------------------
View:
    mov ax,dx
    mov dx,0
    div cx
    call ViewNo
    mov bx,dx
    mov dx,0
    mov ax,cx
    mov cx,10
    div cx
    mov dx,bx
    mov cx,ax
    cmp ax,0
    jne View
    ret

ViewNo:
    push ax
    push dx
    mov dx,ax
    add dl,30h
    mov ah,2
    int 21h
    pop dx
    pop ax
    ret

; ------------------------
; EXIT ROUTINE
; ------------------------
exit:
    mov dx, offset msg6
    mov ah, 09h
    int 21h
```

```
mov ah, 0
int 16h
ret
```