

```

assume cs:code, ds:data

data segment
    msg1 db 0ah, 0dh, "Enter 1st number: $"
    msg2 db 0ah, 0dh, "Enter 2nd number: $"
    msg3 db 0ah, 0dh, "The product is: $"
data ends

code segment

readnum macro msg
    local num, next

    lea dx, msg
    mov ah, 09h
    int 21h
    mov bx, 0

    num:
        mov ah, 01h
        int 21h
        cmp al, 0dh
        jz next
        mov ah, 0
        sub al, 30h
        push ax
        mov ax, 0Ah
        mul bx
        pop bx
        add bx, ax
        jmp num
    next:
        push bx
endm

start:
    mov ax, data
    mov ds, ax

    readnum msg1
    readnum msg2

    pop bx ; num2 is popped from stack into bx
    pop ax ; num is popped from stack into ax
    mul bx ; dx:ax = (ax)*(bx)
    push ax ; push ax (result) into stack

    lea dx, msg3

```

```

mov ah, 09h
int 21h

pop ax      ; read back result from stack
mov cx, 0   ; set counter to 0
mov dx, 0   ; clear dx to 0
mov bx, 0Ah ; set divisor to 10d
divide:
    div bx ; (dx)(ax)/(bx) quotient in ax, remainder in dx
    push dx ; push remainder in dx to stack
    mov dx, 0 ; clear dx
    inc cx ; increment count
    or ax, ax ; repeat till quotient not 0
    jnz divide
print:
    pop dx ; pop the remainder from stack
    add dl, 30h ; convert to ASCII for printing
    mov ah, 02h
    int 21h
    loop print
mov ah, 4ch
int 21h
code ends
end start

```