Addition

```
assume cs:code, ds:data
data segment
    msg1 db 0ah,0dh, "Enter no: $"
    msg2 db 0ah,0dh,"Sum is: $"
    num1 dw 0000h
    num2 dw 0000h
data ends

code segment
    print macro msg
    lea dx,msg
    mov ah,09h
    int 21h
endm

read proc
    xor ax,ax
    push ax
    l1:
        mov ah,01h
        int 21h
        cmp al,0dh
        je l2
        mov ah,00h
        sub al,30h
        mov bx,ax
        mov dx,000ah
        pop ax
        mul dx
        add ax,bx
        push ax
        jmp l1
    l2:
        pop ax
        ret
read endp

display proc
    push dx
    mov bx,000ah
    xor cx,cx
    l3:
        xor dx,dx
        div bx
        add dx,0030h
        push dx
        inc cx
        cmp ax,0000h
        jnz l3
    l4:
        pop dx
        mov ah,02h
        int 21h
        loop l4
        pop dx
        ret
display endp

addtn proc
    mov ax,num1
    mov bx,num2
    add ax,bx
    call display
    ret
addtn endp

start:
    mov ax,data
    mov ds,ax
    print msg1
    call read
    mov num1,ax
    print msg1
    call read
    mov num2,ax
    print msg2
    call addtn
    mov ah,4ch
    int 21h
    code ends
end start
```

Output

```
C:\>add
Enter no: 50
Enter no: 60
Sum is: 110
```

Substraction

```
assume cs:code, ds:data
data segment
    msg1 db 0ah,0dh, "Enter no: $"
    msg2 db 0ah,0dh,"Difference is: $"
    num1 dw 0000h
    num2 dw 0000h
data ends
code segment
    print macro msg
    lea dx,msg
    mov ah,09h
    int 21h
endm

read proc
    xor ax,ax
    push ax
    l1:
        mov ah,01h
        int 21h
        cmp al,0dh
        je l2
        mov ah,00h
        sub al,30h
        mov bx,ax
        mov dx,000ah
        pop ax
        mul dx
        add ax,bx
        push ax
        jmp l1
    l2:
        pop ax
        ret
read endp
```

```
display proc
    push dx
    mov bx,000ah
    xor cx,cx
    l3:
        xor dx,dx
        div bx
        add dx,0030h
        push dx
        inc cx
        cmp ax,0000h
        jnz l3
    l4:
        pop dx
        mov ah,02h
        int 21h
        loop l4
        pop dx
        ret
display endp

subtract proc
    mov ax,num1
    mov bx,num2
    sub ax,bx
    call display
    ret
subtract endp

start:
    mov ax,data
    mov ds,ax
    print msg1
    call read
    mov num1,ax
    print msg1
    call read
    mov num2,ax
    print msg2
    call subtract
    mov ah,4ch
    int 21h
    code ends
end start
```

Output

```
C:\>sub
Enter no: 50
Enter no: 30
Difference is: 20
```

Multiplication

```
assume cs:code, ds:data
data segment
    msg1 db 0ah,0dh, "Enter no: $"
    msg2 db 0ah,0dh,"Product is: $"
    num1 dw 0000h
    num2 dw 0000h
data ends
code segment
    print macro msg
    lea dx,msg
    mov ah,09h
    int 21h
endm

read proc
    xor ax,ax
    push ax
    l1:
        mov ah,01h
        int 21h
        cmp al,0dh
        je l2
        mov ah,00h
        sub al,30h
        mov bx,ax
        mov dx,000ah
        pop ax
        mul dx
        add ax,bx
        push ax
        jmp l1
    l2:
        pop ax
        ret
read endp
```

```
display proc
    push dx
    mov bx,000ah
    xor cx,cx
    l3:
        xor dx,dx
        div bx
        add dx,0030h
        push dx
        inc cx
        cmp ax,0000h
        jnz l3
    l4:
        pop dx
        mov ah,02h
        int 21h
        loop l4
        pop dx
        ret
display endp

multiply proc
    mov ax,num1
    mov bx,num2
    mul bx
    call display
    ret
multiply endp

start:
    mov ax,data
    mov ds,ax
    print msg1
    call read
    mov num1,ax
    print msg1
    call read
    mov num2,ax
    print msg2
    call multiply
    mov ah,4ch
    int 21h
    code ends
end start
```

Output

```
C:\>mul
Enter no: 20
Enter no: 30
Product is: 600
```

Division

```
assume cs:code, ds:data
data segment
    msg1 db 0ah,0dh, "Enter no: $"
    msg2 db 0ah,0dh,"Quotient is: $"
    num1 dw 0000h
    num2 dw 0000h
data ends
code segment
    print macro msg
    lea dx,msg
    mov ah,09h
    int 21h
endm

read proc
    xor ax,ax
    push ax
    l1:
        mov ah,01h
        int 21h
        cmp al,0dh
        je l2
        mov ah,00h
        sub al,30h
        mov bx,ax
        mov dx,000ah
        pop ax
        mul dx
        add ax,bx
        push ax
        jmp l1
    l2:
        pop ax
        ret
read endp

display proc
    push dx
    mov bx,000ah
    xor cx,cx
    l3:
        xor dx,dx
        div bx
        add dx,0030h
        push dx
        inc cx
        cmp ax,0000h
        jnz l3
    l4:
        pop dx
        mov ah,02h
        int 21h
        loop l4
        pop dx
        ret
display endp
divide proc
    mov ax,num1
    mov bx,num2
    xor dx,dx
    div bx
    call display
    ret
divide endp

start:
    mov ax,data
    mov ds,ax
    print msg1
    call read
    mov num1,ax
    print msg1
    call read
    mov num2,ax
    print msg2
    call divide
    mov ah,4ch
    int 21h
    code ends
end start
```

Output

```
C:\>div
Enter no: 60
Enter no: 5
Quotient is: 12
```

Factorial

```
ASSUME CS:code, DS:data
data SEGMENT
    msg1 DB 0Ah, 0Dh, "Enter no: $"
    msg2 DB 0Ah, 0Dh, "Factorial is: $"
    num1 DW 0000h
data ENDS

code SEGMENT
    print MACRO msg
    lea dx, msg
    mov ah, 09h
    int 21h
ENDM

read PROC
    xor ax, ax
    push ax
    l1:
        mov ah, 01h
        int 21h
        cmp al, 0Dh
        je l2
        mov ah, 00h
        sub al, 30h
        mov bx, ax
        mov dx, 000Ah
        pop ax
        mul dx
        add ax, bx
        push ax
        jmp l1
    l2:
        pop ax
        ret
read ENDP
```

```
display PROC
    push dx
    mov bx, 000Ah
    xor cx, cx
    l3:
        xor dx, dx
        div bx
        add dx, 0030h
        push dx
        inc cx
        cmp ax, 0000h
        jnz l3
    l4:
        pop dx
        mov ah, 02h
        int 21h
        loop l4
        pop dx
        ret
display ENDP

factorial PROC
    mov ax, 0001h
    mov cx, num1
    l5:
        mul cx
        dec cx
        jnz l5
        call display
        ret
factorial ENDP

start:
    mov ax, data
    mov ds, ax
    print msg1
    call read
    mov num1, ax
    print msg2
    call factorial
    mov ah, 4Ch
    int 21h
    code ENDS
END start
```

Output

```
C:\>fact
Enter no: 6
Factorial is: 720
```