	13:
Addition	xor dx , dx
Addition	div bx
assume cs :code, ds :data	add dx ,0030h
data segment	push dx
msg1 db 0ah,0dh, "Enter no: \$"	inc cx
msg2 db 0ah,0dh,"Sum is: \$"	cmp ax ,0000h
num1 <mark>dw</mark> 0000h	jnz l3
num2 <mark>dw</mark> 0000h	l4:
data ends	pop dx
	mov ah ,02h
code segment	int 21h
print macro msg	loop l4
lea dx , msg	pop dx
mov ah ,09h	ret
int 21h	display endp
endm	
	addtn proc
read proc	mov ax ,num1
xor ax , ax	mov bx ,num2
push ax	add ax,bx
l1:	call display
mov ah ,01h	ret
int 21h	addtn endp
cmp al ,0dh	
je l2	start:
mov ah ,00h	mov ax ,data
sub al ,30h	mov ds , ax
mov b x, a x	print msg1
mov dx ,000ah	call read
pop ax	mov num1,ax
mul dx	print msg1
add ax, bx	call read
push ax	mov num2, ax
jmp l1	print msg2
l2:	call addtn
pop ax	mov ah ,4ch
ret	int 21h
read endp	code ends
	end start
display proc	
push dx	
mov bx ,000ah	
xor cx, cx	

```
C:\>add
Enter no: 50
Enter no: 60
Sum is: 110
Substraction
assume cs:code, ds:data
data segment
   msg1 db Oah, Odh, "Enter no: $"
   msg2 db Oah, Odh, "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
    12:
        pop ax
        ret
read endp
```

Output

display proc	Output	display proc
push dx		push dx
mov bx ,000ah	C:\>sub	mov bx ,000ah
xor cx, cx	Enter no: 50	xor cx,cx
13:	Enter no: 30	l3:
xor dx , dx	Difference is: 20	xor dx , dx
div bx		div bx
add dx ,0030h	Multiplication	add dx ,0030h
push dx	Multiplication	push dx
inc cx	assume cs :code, ds :data	inc cx
cmp ax ,0000h	data segment '	cmp ax ,0000h
jnz 13	msg1 db Oah,Odh, "Enter no: \$"	jnz la
14:	msg2 db Oah,Odh,"Product is: \$"	14:
pop dx	num1 dw 0000h	pop dx
mov ah ,02h	num2 dw 0000h	mov ah ,02h
int 21h	data ends	int 21h
loop 14	code segment	loop 14
pop dx	print macro msq	pop dx
ret	lea dx , msg	ret
display endp	mov ah, 09h	display endp
also tay shap	int 21h	disp tay thup
subtract proc	endm	multiply proc
mov ax, num1	C.T.G.III	mov ax, num1
mov bx, num2	read proc	mov bx, num2
sub ax , bx	xor ax, ax	mul bx
call display	push ax	call display
ret	l1:	ret
subtract endp	mov ah , 01h	multiply endp
ouser doe on up	int 21h	materpty onap
start:	cmp al, 0dh	start:
mov ax, data	je l2	mov ax , data
mov ds, ax	mov ah ,00h	mov ds, ax
print msg1	sub al , 30h	print msg1
call read	mov bx, ax	call read
mov num1, ax	mov dx ,000ah	mov num1,ax
print msg1	pop ax	print msg1
call read	mul dx	call read
mov num2, ax	add ax , bx	mov num2, ax
print msg2	push ax	print msg2
call subtract	jmp l1	call multiply
mov ah, 4ch	l2:	mov ah , 4ch
int 21h	pop ax	int 21h
code ends	ret	code ends
end start	read endp	end start
<u> </u>	. 044 01145	-110 OCAL C

Output	display proc	Output
0.15	push dx	٠٠١٠ الله
C:\>mul	mo∨ bx ,000ah	C:\>div
Enter no: 20	xor cx, cx	Enter no: 60
Enter no: 30	13:	Enter no: 5
Product is: 600	xor dx , dx	Quotient is: 12
	div b x	
Division	add dx ,0030h	Factorial
DIVISION	push dx	lactorial
assume cs :code, ds :data	inc cx	ASSUME CS :code, DS :data
data segment	cmp ax ,0000h	data SEGMENT
msg1 db Oah,Odh, "Enter no: \$"	jnz l3	msg1 DB OAh, ODh, "Enter no: \$"
msg2 db Oah, Odh, "Quotient is: \$"	14:	msg2 DB OAh, ODh, "Factorial is: \$"
num1 dw 0000h	pop d x	num1 DW 0000h
num2 dw 0000h	mov ah , 02h	data ENDS
data ends	int 21h	data Endo
code segment	loop 14	code SEGMENT
print macro msg	pop dx	print MACRO msg
lea dx , msg	ret	lea dx , msg
mov ah , 09h	display endp	mov ah, 09h
int 21h	divide proc	int 21h
	·	
endm	mov ax, num1	ENDM
wood nano	mov bx, num2	mood DDOC
read proc	xor dx, dx	read PROC
xor ax, ax	div bx	xor ax, ax
push ax	call display	push ax
l1:	ret	l1:
mov ah , 01h	divide endp	mov ah , 01h
int 21h		int 21h
cmp al ,0dh	start:	cmp al , 0D h
je l2	mov ax ,data	je l2
mov ah ,00h	mov ds , ax	mo∨ ah , 00h
sub al ,30h	print msg1	sub al , 30h
mov b x, a x	call read	mo∨ bx , ax
mov dx ,000ah	mo∨ num1,ax	mo∨ dx, 000 <mark>Ah</mark>
pop ax	print msg1	pop ax
mul dx	call read	mul dx
add ax,bx	mo∨ num2, ax	add ax , bx
push ax	print msg2	push ax
jmp l1	call divide	jmp l1
12:	mov ah , 4ch	l2:
pop ax	int 21h	pop ax
ret	code ends	ret
read endp	end start	read ENDP
· · · · · · ·		· • • • · · · · · · · · · · · · · · · ·

```
display PROC
                                            Output
   push dx
   mov bx, 000Ah
                                            C:\>fact
   xor cx, cx
                                            Enter no: 6
                                            Factorial is: 720
   13:
       xor dx, dx
       div bx
       add dx, 0030h
       push dx
       inc cx
       cmp ax, 0000h
       jnz l3
   14:
       pop dx
       mov ah, 02h
       int 21h
       loop 14
       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
```