SYSTEM PROGRAMMING LAB

NAME: MU IN NASIF
BCSE III sem 5
001910501036
ASSIGNMENT 2

1. Write and test a MASM program to add and substract two 16bit numbers

__CODE:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                  ;;
;;              addition                            ;;
;;                                                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

dosseg
.model small
.stack 100h
.data
msg1 db 10,13,'Enter 1st number:$'
msg2 db 10,13,'Enter 2nd number:$'
msg3 db 10,13,'Sum is $'

.code
main proc

mov ax,@data
mov ds,ax

lea dx,msg1
mov ah,09h
int 21h

mov ah,01h
int 21h

mov bl,al
lea dx,msg2
mov ah,09h
int 21h

mov ah,01h
int 21h
mov ch,al                ;second number first digit

mov ah,01h
int 21h
mov cl,al                ;second number second digit
                         ;both number input taking is done.
                         ;now we start computing lower digit addition first
```

```
mov al,bl
mov ah,00h              ;clear ah first
add al,cl               ;al=al+cl, if carry generated then it is stored in ah
aaa

add ax,3030h            ;convert to ascii code
mov bl,al               ;store lower digit sum of the result

                        ;now we start computing higher digit addition alongwith carry 'ah'

mov al,ah               ;store carry
mov ah,00h
add al,bh               ;al+=bl
add al,ch               ;al+=cl
aaa

add ax,3030h            ;convert to ascii code
mov bh,al               ;store al in bh;store higher digit sum of the result

mov cl,ah               ;store final carry in cl

lea dx, msg3
mov ah,09h
int 21h

mov dl,cl
                        ;print carry
mov ah,02h
int 21h
                        ;print higher digit
mov dl,bh
mov ah,02h
int 21h
                        ;print lower digit
mov dl,bl
mov ah,02h
int 21h

mov ah,4ch
int 21h

main endp
end main

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                      ;;
;; substraction                         ;;
;;                                      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

dosseg
```

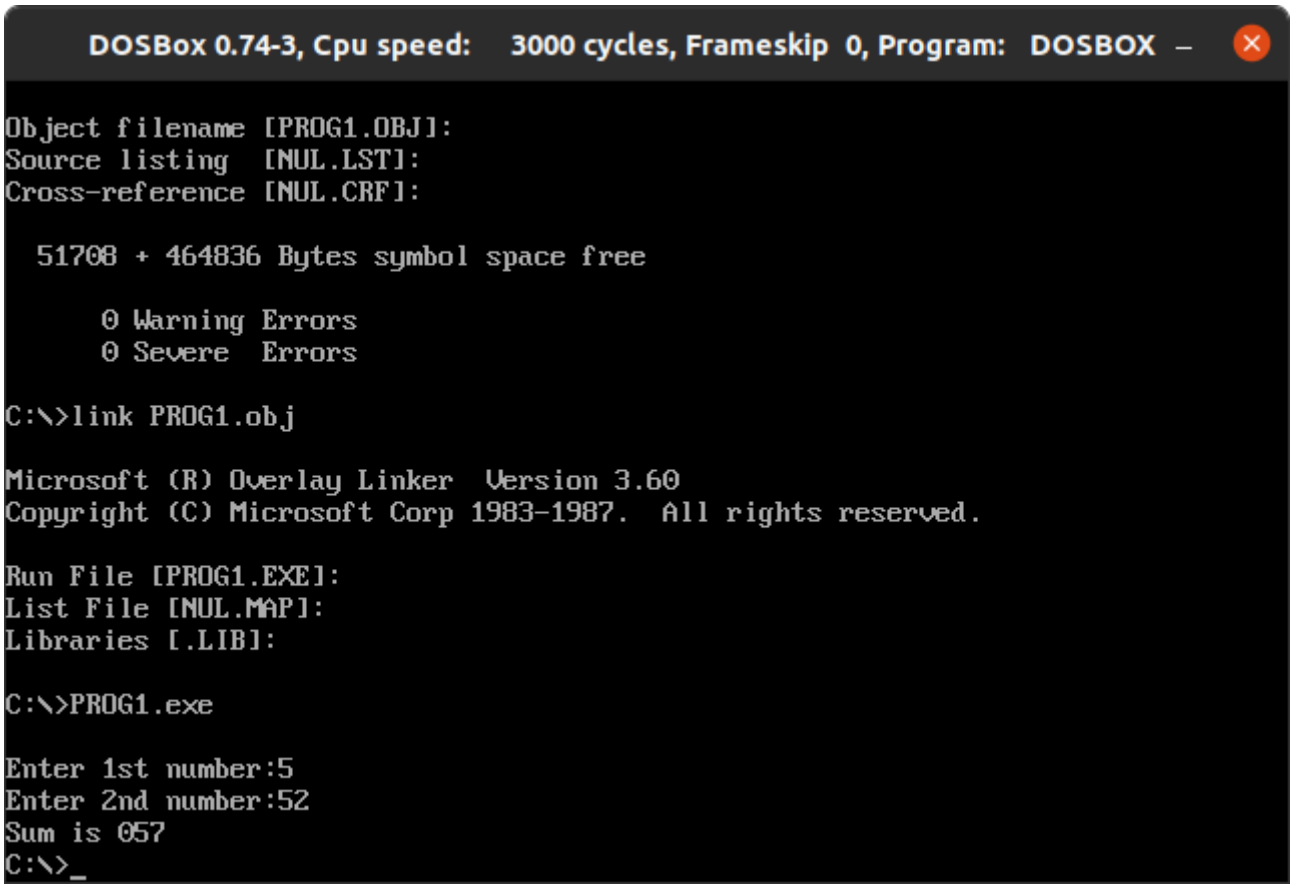
```
.model small
.stack 100h
.data
msg1 db 10,13,'Enter 1st number: $'
msg2 db 10,13,'Enter 2nd number :$'
msg3 db 10,13,'Difference is $'
msg4 db 10,13,'Difference is negative$'
.code
main proc

                        ;initialize the data
mov ax,@data
mov ds,ax

lea dx,msg1
mov ah,09h
int 21h
                        ;input first number
mov ah,01h
int 21h
mov bh,al               ;first number first digit

mov ah,01h
int 21h
mov bl,al               ;first number second digit

lea dx,msg2
mov ah,09h
int 21h
                        ;second first number
mov ah,01h
int 21h
mov ch,al               ;second number first digit

mov ah,01h
int 21h
mov cl,al               ;second number second digit
                        ;both number input taking is done.

cmp cx,bx               ;compare the diffrence
jz ZERO                     ;if both are same then zero flag=1 detected then jump to ZERO
label
jnc NEGATIVE                ;if source is greater than destiantion jump to NEGATIVE label
clc                     ;clear carry flag i.e. difference is positive hence carry reset to be done
                        ;for later calculation
mov al,bl
mov ah,00
sub al,cl               ;al=al-cl
aas
mov cl,ah               ;store carry
add al,30h              ;convert to ascii code
mov bl,al               ;bl=al ;store lower digit of the result
```

```
mov ah,00
sub bh,ch                ;bh=bh-ch
mov al,bh                ;al=bh
aas
add al,cl                ;al=al+cl
add al,30h               ;convert ascii code
mov bh,al                ;store higher digit of the result
jmp DISP

ZERO:
lea dx,msg3
mov ah,9
int 21h
mov ax,bx                ;move bx into ax
sub ax,cx                ;ax=ax-cx
mov bh,ah                ;bh=ah
add al,30                ;convert to ascii
mov dl,al
mov ah,2
int 21h
add bh,30                ;convert to ascii
mov dl,bh
mov ah,2
int 21h
jmp FULL

NEGATIVE:
mov al,cl
mov ah,00
sub al,bl
aas
mov bl,ah
add al,30
mov cl,al
mov ah,00
sub ch,bh
mov al,ch
aas
add al,bl
add al,30
mov ch,al
lea dx, msg4
mov ah,9
int 21h
mov dl,ch
mov ah,02h
int 21h                  ;print higher digit
mov dl,cl
mov ah,2
int 21h
jmp FULL
```

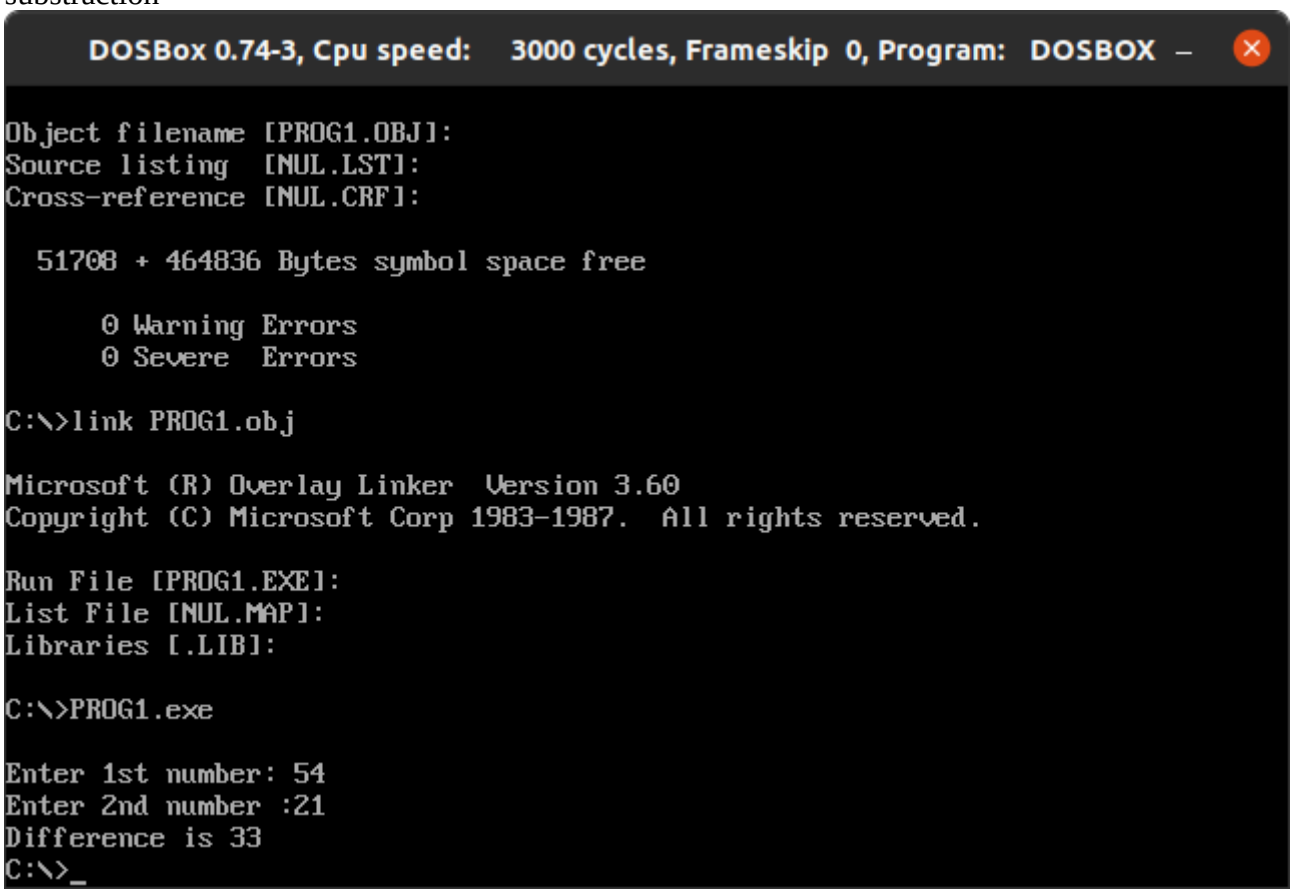
```
DISP:
lea dx,msg3
mov ah,9
int 21h
mov dl,bh
mov ah,2
int 21h                 ;print higher digit
mov dl,bl
mov ah,2
int 21h                 ;print lower digit



FULL:
mov ah,4ch
int 21h



main endp
end main
```

__OUTPUT:

addition



substraction



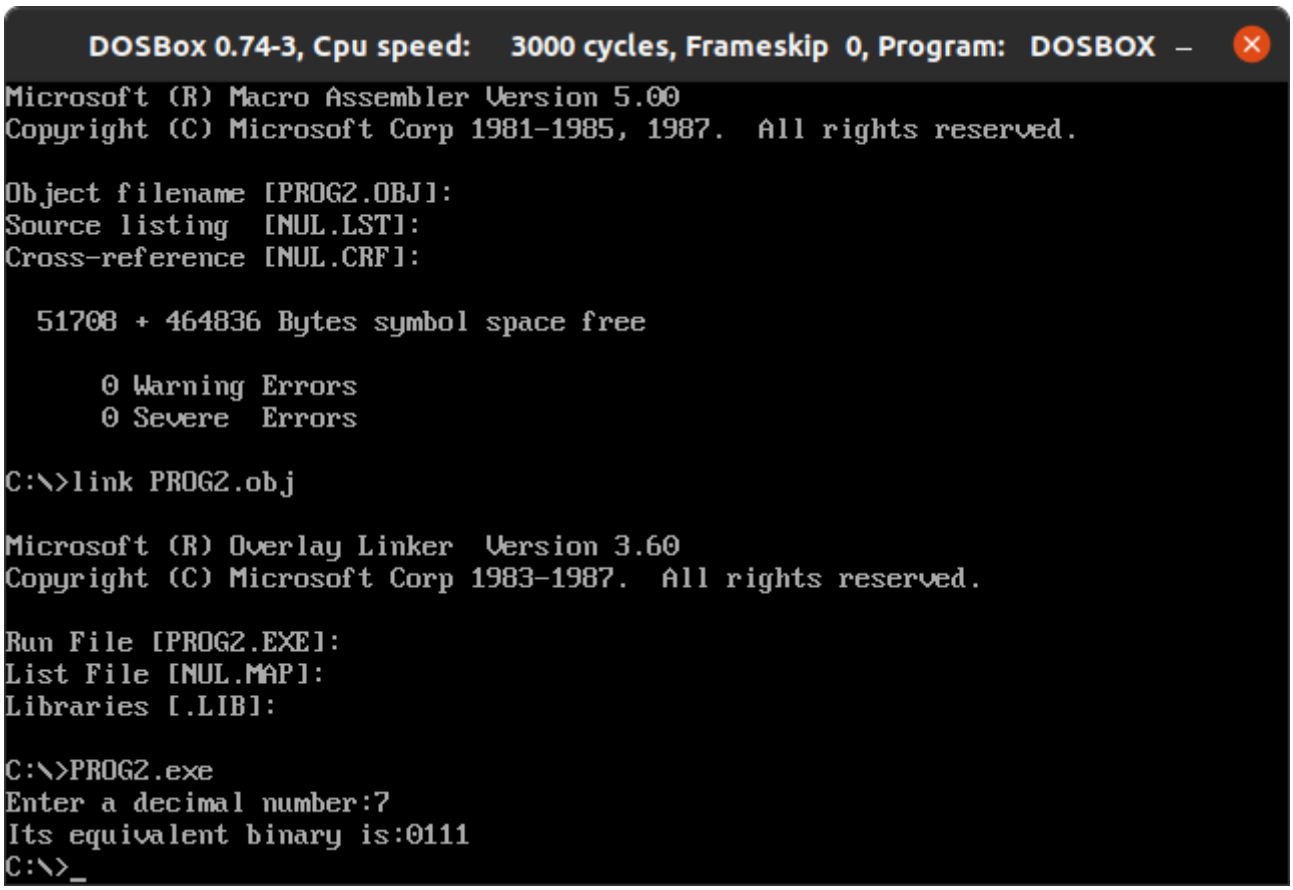
2. Write and test a MASM program to convert Binary digit to Decimal and vice versa

__CODE:

```
.model small
.stack 100h
.data
msg db 'Enter a decimal number:$'
msg1 db 0dh,0ah,'Invalid entry $'
msg2 db 0dh,0ah,'Its equivalent binary is:$'
.code
main proc
mov ax,@data
mov ds,ax
lea dx,msg
mov ah,9 ;print message
int 21h
mov ah,1
int 21h ;read data from user
cmp al,30h ;check whether user enter number or something else
jnge invalid ;jump if any invalid entry
cmp al,39h
jnle invalid
lea dx,msg2 ;print message
mov ah,9
int 21h
and al,0fh ;clear upper four bits of al register
mov cl,3 ;cl used as counter in shifting bits
mov bl,al ;save value in bl register
mov bh,bl ;move contents of bl into bh
shr bh,cl ;right shift bh register three times by using cl as a counter
add bh,30h ;make binary value visible as 0 or 1
mov ah,2 ;print binary value
mov dl,bh
int 21h
xor bh,bh ;clear bh register
mov bh,bl
mov cl,2 ;make cl counter value equals to two
and bh,04h ;clear all bits except third last bit
shr bh,cl
add bh,30h
mov ah,2 ;print binary value of third last bit
mov dl,bh
int 21h
xor bh,bh
mov bh,bl
and bh,02h ;clear all bits except second last bit
shr bh,1
add bh,30h
mov ah,2 ;print second last bit
```

```
mov dl,bh
int 21h
xor bh,bh
mov bh,bl
and bh,01h ;clear all bits except the last bit
add bh,30h
mov ah,2 ;print last bit in binary
mov dl,bh
int 21h
jmp exit
invalid:
lea dx,msg1 ;used to print message of invalid entry
mov ah,9
int 21h
exit:
mov ah,4ch
int 21h
main endp
end main
```

__OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [PROG2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51708 + 464836 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link PROG2.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [PROG2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
C:\>PROG2.exe
Enter a decimal number:7
Its equivalent binary is:0111
C:\>_

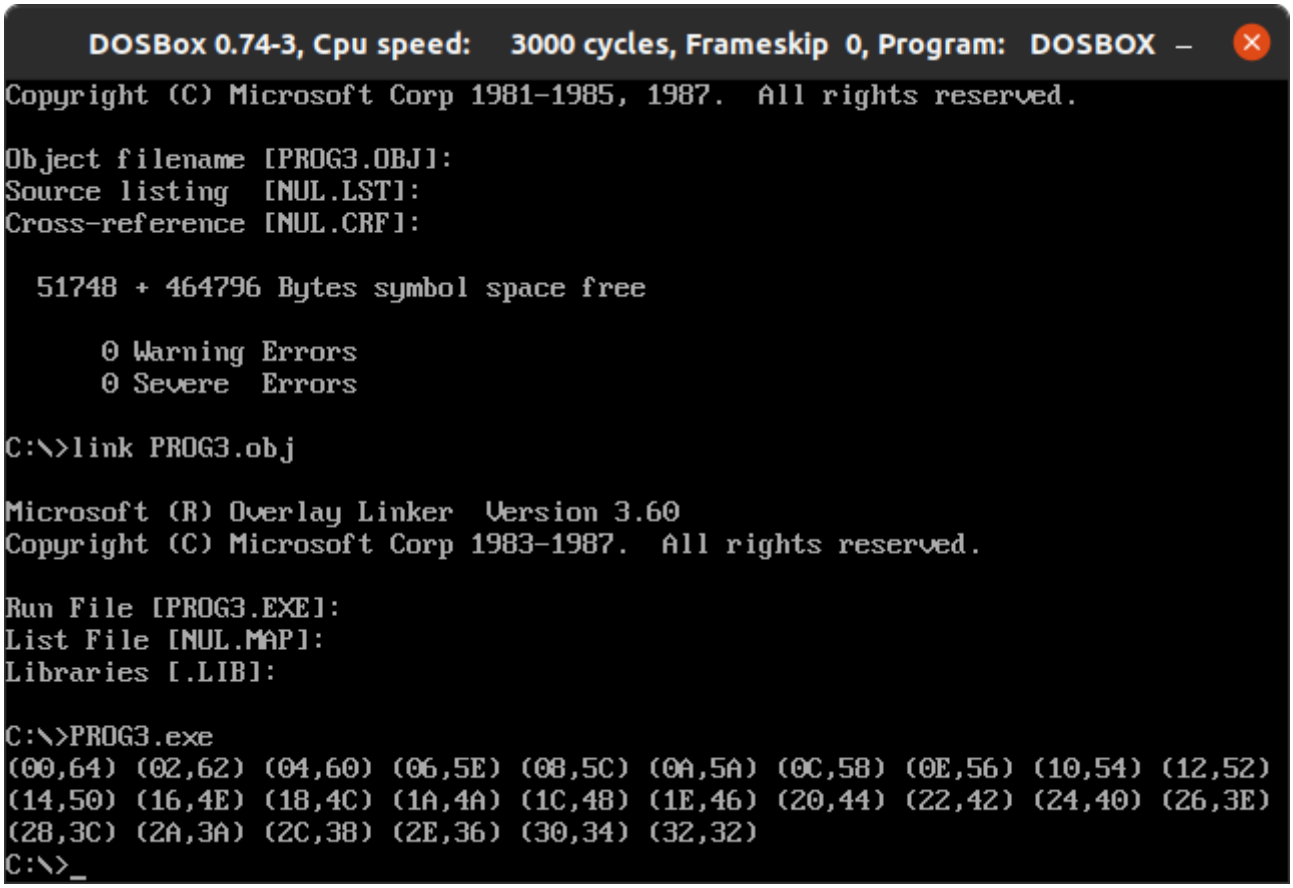
3. Write and test a program to print pairs of even numbers where the summation of the numbers in each pair is 100

__CODE:

```
.model small
.stack 100h

.data
num DB ?

.code
mov ax,@data
mov ds,ax
call main
mov ah,4ch
int 21h

main proc
    ;initialize al with 0
    mov bl,0
    mov bh,100

    loop1:

    ;output

    ;display opening braces
    mov dl,40
    mov ah,2
    int 21h

    ;display number 1
    mov num,bl
    call outputNum

    ;display coma
    mov dl,44
    mov ah,2
    int 21h

    ;display number 2
    mov num,bh
    call outputNum

    ;display closing braces
    mov dl,41
    mov ah,2
    int 21h

    ;display space
    mov dl,32
```

```
    mov ah,2
    int 21h

    ;increment al
    inc bl
    inc bl

    ;decrement bh
    dec bh
    dec bh

    ;compare al with 51
    cmp bl,50
    jle loop1
    ret
main endp

; program to output a number stored in num
outputNum proc
    push cx
    push dx
    push ax

    mov cl, 4
    mov dl, num
    shr dl,cl
    and dl, 0fh
    cmp dl,0ah
    jl isNumber2
    add dl,7
    isNumber2:
    add dl,48
    mov ah, 2
    int 21h

    mov dl, num
    and dl, 0fh
    cmp dl, 0ah
    jl isNumber1
    add dl, 07h
    isNumber1:
    add dl,48
    mov ah, 2
    int 21h

    pop ax
    pop dx
    pop cx
    ret
outputNum endp

end
```

__OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [PROG3.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51748 + 464796 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link PROG3.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [PROG3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
C:\>PROG3.exe
(00,64) (02,62) (04,60) (06,5E) (08,5C) (0A,5A) (0C,58) (0E,56) (10,54) (12,52)
(14,50) (16,4E) (18,4C) (1A,4A) (1C,48) (1E,46) (20,44) (22,42) (24,40) (26,3E)
(28,3C) (2A,3A) (2C,38) (2E,36) (30,34) (32,32)
C:\>

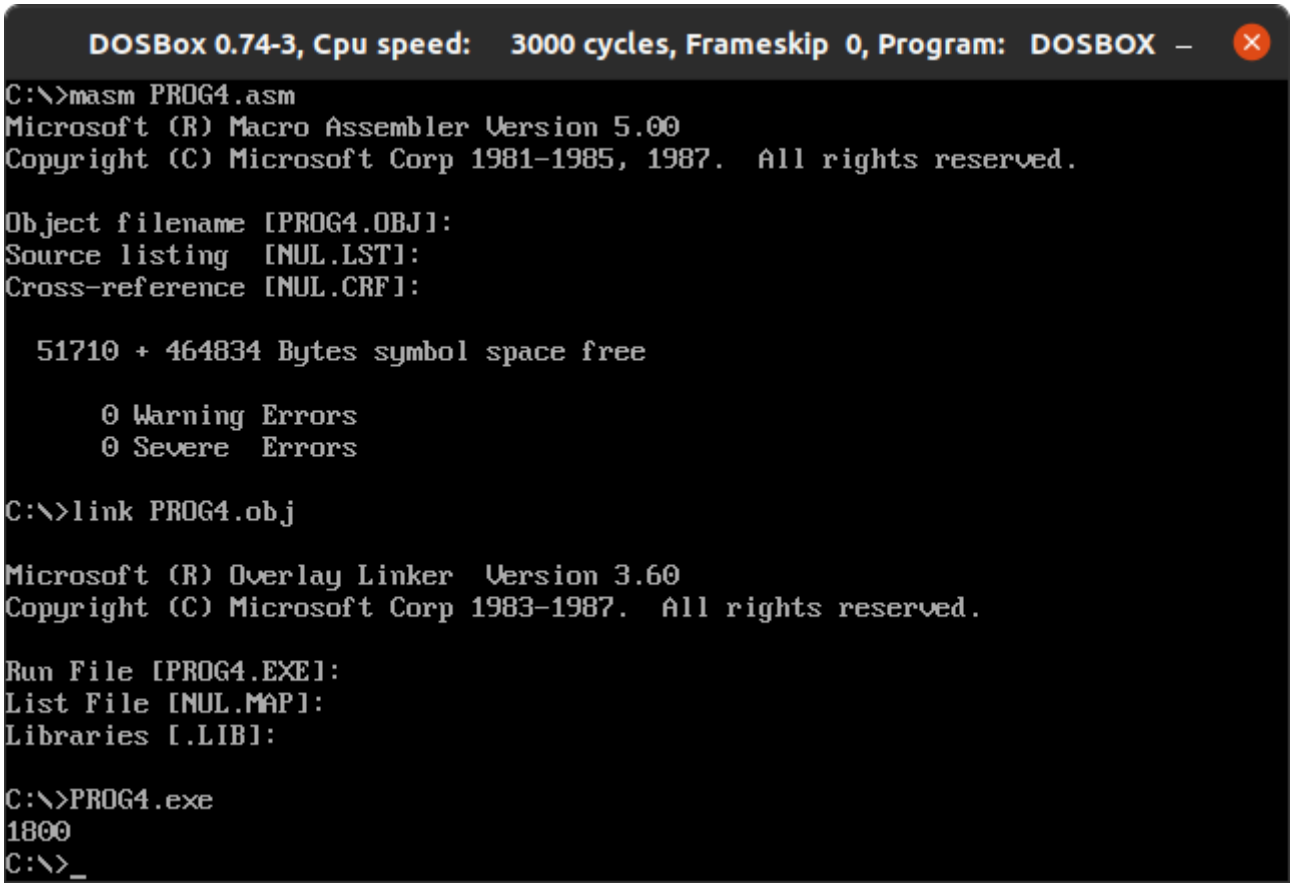
4. Write and test a MASM program to multiply two 32bit numbers

__CODE:

```
dosseg
.model small
.stack 100h
.data
.code
MAIN proc
mov ax,20
mov cx,90
mul cx
CALL PRINT

MOV AH, 4CH
INT 21H
MAIN endp
PRINT PROC

    ;initialize count
    mov cx,0
    mov dx,0
    label1:
        ; if ax is zero
        cmp ax,0
        je print1

        ;initialize bx to 10
        mov bx,10

        ; extract the last digit
        div bx

        ;push it in the stack
        push dx

        ;increment the count
        inc cx

        ;set dx to 0
        xor dx,dx
        jmp label1
    print1:
        ;check if count
        ;is greater than zero
        cmp cx,0
        je exit

        ;pop the top of stack
        pop dx
```

```
        ;add 48 so that it
        ;represents the ASCII
        ;value of digits
        add dx,48

        ;interrupt to print a
        ;character
        mov ah,02h
        int 21h

        ;decrease the count
        dec cx
        jmp print1
exit:
ret
PRINT ENDP


End MAIN
```

__OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
C:\>masm PROG4.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [PROG4.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51710 + 464834 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link PROG4.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [PROG4.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
C:\>PROG4.exe
1800
C:\>_

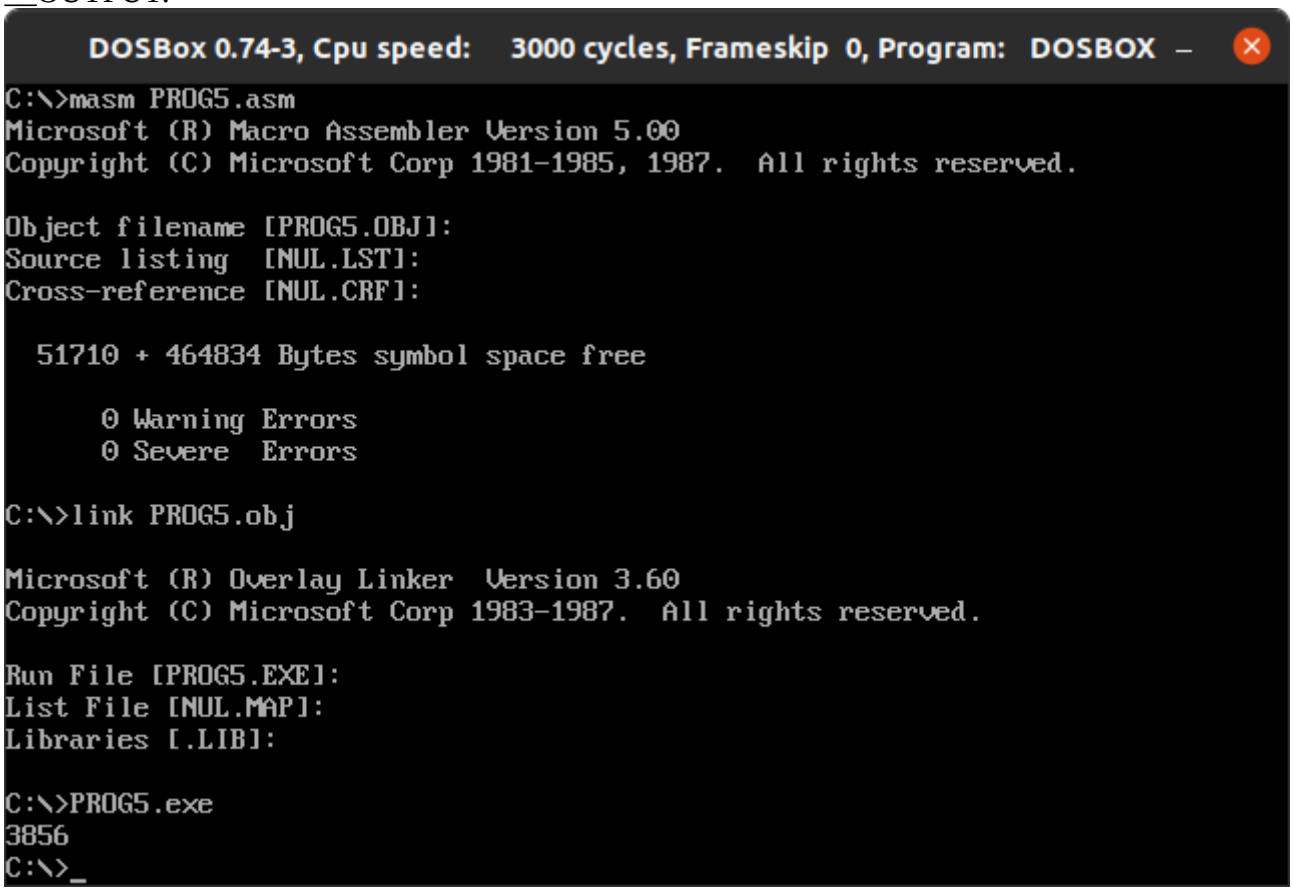
5. Write and test a MASM program to divide a 16bit number by an 8bit number

__CODE:

```
dosseg
.model small
.stack 100h
.data
.code
MAIN proc
mov dx,0
mov ax,3258
mov cx,85
div cx
CALL PRINT


mov ax,dx
CALL PRINT
MOV AH, 4CH
INT 21H
MAIN endp
PRINT PROC

    ;initialize count
    mov cx,0
    mov dx,0
    label1:
        ; if ax is zero
        cmp ax,0
        je print1

        ;initialize bx to 10
        mov bx,10

        ; extract the last digit
        div bx

        ;push it in the stack
        push dx

        ;increment the count
        inc cx

        ;set dx to 0
        xor dx,dx
        jmp label1
    print1:
        ;check if count
        ;is greater than zero
        cmp cx,0
```

```
        je exit

        ;pop the top of stack
        pop dx

        ;add 48 so that it
        ;represents the ASCII
        ;value of digits
        add dx,48

        ;interrupt to print a
        ;character
        mov ah,02h
        int 21h

        ;decrease the count
        dec cx
        jmp print1
exit:
ret
PRINT ENDP


End MAIN
```

__OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
C:\>masm PROG5.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [PROG5.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51710 + 464834 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\>link PROG5.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [PROG5.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
C:\>PROG5.exe
3856
C:\>_