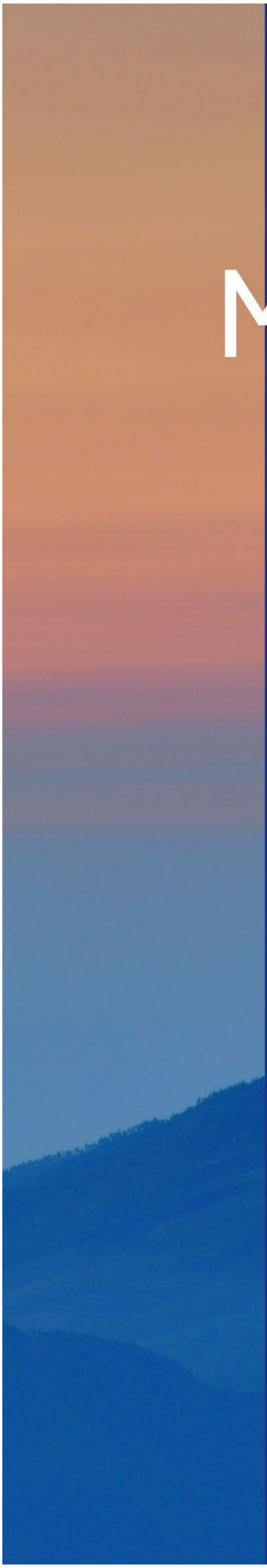


Microprocessor

ASSIGNMENT - 1

SUBMITTED BY:

ABHISHEK SAHU
18012

SUBMITTED TO:

Mr Mahesh Bhandari

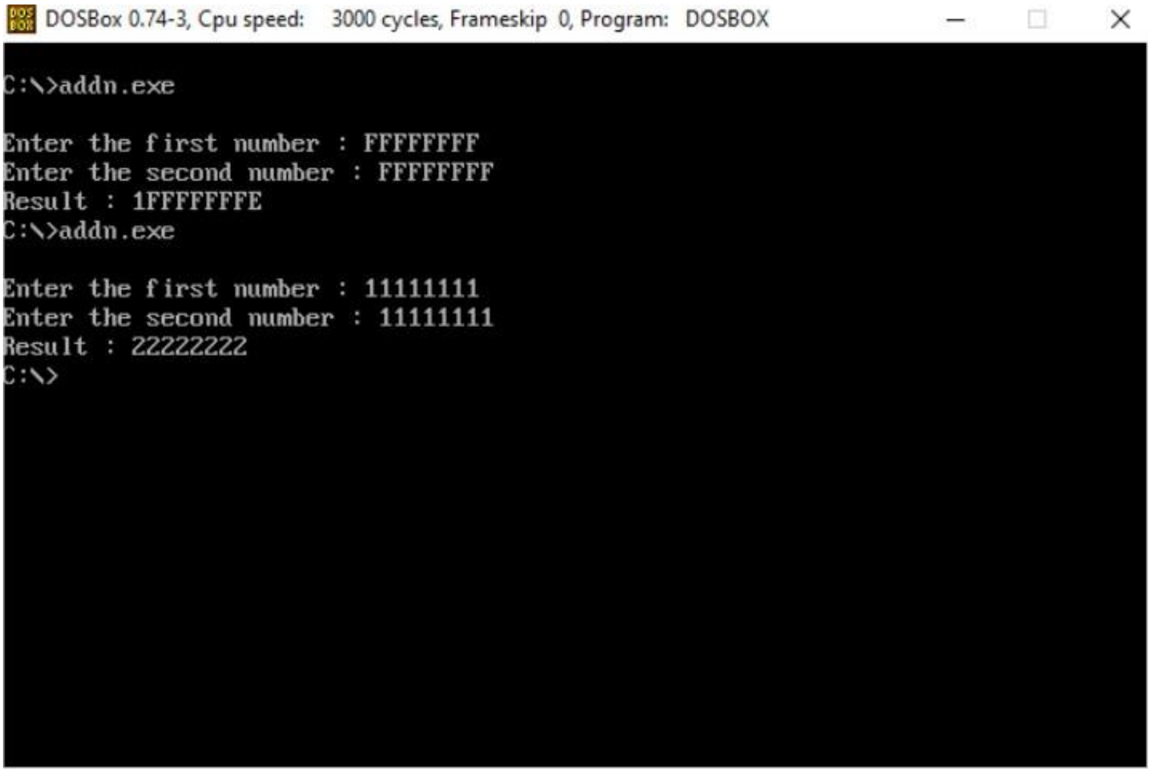
Q1. Write a program for 32 bit binary addition, subtraction, multiplication and division.

1. Addition Code:

```
.model small
.486
.data
str1 db 10,13, "Enter the first number : $"
str2 db 10,13, "Enter the second number : $"
str3 db 10,13, "Result : $"
num1 db 4 dup(0)
num2 db 4 dup(0)
.stack 200h
.code
.startup

; Procedures Used ->
; prints  : prints the string terminated by $
; printd : prints the result
; getVal1 : to input num1
; getVal2 : to input num2

lea dx,str1
call prints
call getVal1
lea dx,str2
call prints
call getVal2
mov si,3
mov cx,4

;addition
clc ;clear CF
addn:
  mov bl,num2[si]
  adc num1[si],bl
  dec si
loop addn

lea dx,str3
```

```
call prints
call printd
.exit

prints proc near
        mov ah,9
        int 21h
        ret
prints endp

getVal1 proc near
        mov  si,0
        mov cx,4
        ab1:
                mov ah,1
                int 21h
                sub al,30h
           .if al > 9h
                   sub al,7h
           .endif
                mov num1[si],al
                shl num1[si],4
                mov ah,1
                int 21h
                sub al,30h
                .if al > 9h
                        sub al,7h
                .endif
                add num1[si],al
                inc si
        loop ab1
        ret
getVal1 endp

getVal2 proc near
        mov  si,0
        mov cx,4
        ab2:
                mov ah,1
                int 21h
                sub al,30h
           .if al > 9h
                   sub al,7h
           .endif
                mov num2[si],al
                shl num2[si],4
                mov ah,1
                int 21h
                sub al,30h
                .if al > 9h
                        sub al,7h
                .endif
                add num2[si],al
                inc si
        loop ab2
        ret
getVal2 endp

printd proc near
        mov  si,0
        mov cx,4
        jnc ab
        mov dl,31h
        mov ah,2
        int 21h
        ab:
                mov dl,num1[si]
                and dl,0f0h
                shr dl,4
                add dl,30h
                .if dl > 39h
                        add dl,7h
                .endif
                mov ah,2
                int 21h
                mov dl,num1[si]
                and dl,0fh
                add dl,30h
                .if dl > 39h
                        add dl,7h
                .endif
                mov ah,2
                int 21h
                inc si
        loop ab
        ret
printd endp

end
```

OUTPUT:



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>addn.exe

Enter the first number : FFFFFFFF
Enter the second number : FFFFFFFF
Result : 1FFFFFFFE
C:\>addn.exe

Enter the first number : 11111111
Enter the second number : 11111111
Result : 22222222
C:\>
```

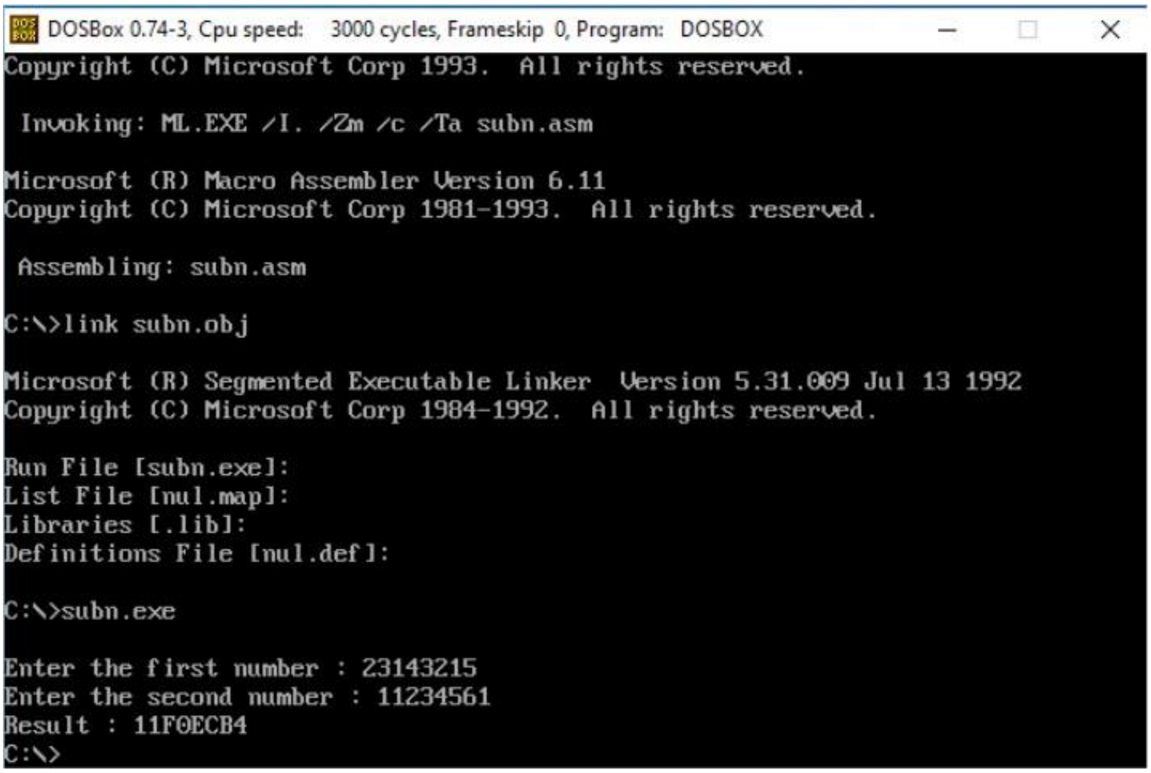
2.Subtraction Code:

```
.model small
.486
.data
str1 db 10,13, "Enter the first number : $"
str2 db 10,13, "Enter the second number : $"
str3 db 10,13, "Result : $"
num1 db 4 dup(0)
num2 db 4 dup(0)
.stack 200h
.code
.startup

; Procedures Used ->
; prints  : prints the string terminated by $
; printd : prints the result
; getVal1 : to input num1
; getVal2 : to input num2

lea dx,str1
call prints
call getVal1
lea dx,str2
call prints
call getVal2
mov si,3
mov cx,4

;subtraction
clc ;clear C Flag
subn:
  mov bl,num2[si]
  sbb num1[si],bl
  dec si
loop subn

lea dx,str3
call  prints
call printd
```

```
.exit

prints proc near
    mov ah,9
    int 21h
    ret
prints endp

getVal1 proc near
    mov  si,0
    mov cx,4
    ab1:
        mov ah,1
        int 21h
        sub al,30h
     .if al > 9h
         sub al,7h
     .endif
        mov num1[si],al
        shl num1[si],4
        mov ah,1
        int 21h
        sub al,30h
        .if al > 9h
            sub al,7h
        .endif
        add num1[si],al
        inc si
    loop ab1
    ret
getVal1 endp

getVal2 proc near
    mov  si,0
    mov cx,4
    ab2:
        mov ah,1
        int 21h
        sub al,30h
     .if al > 9h
         sub al,7h
     .endif
        mov num2[si],al
        shl num2[si],4
        mov ah,1
        int 21h
        sub al,30h
        .if al > 9h
            sub al,7h
        .endif
        add num2[si],al
        inc si
    loop ab2
    ret
getVal2 endp

printd proc near
    mov si,0
    mov cx,4

    ab:
        mov dl,num1[si]
        and dl,0f0h
        shr dl,4
        add dl,30h
        .if dl > 39h
            add dl,7h
        .endif
        mov ah,2
        int 21h
        mov dl,num1[si]
        and dl,0fh
        add dl,30h
        .if dl > 39h
            add dl,7h
        .endif
        mov ah,2
        int 21h
        inc si
    loop ab
    ret
printd endp
```

end

OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta subn.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: subn.asm

C:\>link subn.obj

Microsoft (R) Segmented Executable Linker  Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [subn.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

C:\>subn.exe

Enter the first number : 23143215
Enter the second number : 11234561
Result : 11F0ECB4
C:\>
```

3.Multiplication code:

```
.model small
.stack
.486
.data

ad dd ?
ad1 dd ?
msg db 13,10,"Enter the first number(32 Bit): $"
msg1 db 13,10,"Enter the second number(32 Bit): $"
msg2 db 13,10,"Product: $"

.code
.startup
;input of 32 bit no.
mov dx,offset msg
mov ah,09
int 21h
mov ebx,0
mov cx,4
abc: shl ebx,8
;1st DIGIT OF FIRST NO.
mov ah,01
int 21h
cmp al,39h
jbe ab1
sub al,37h
```

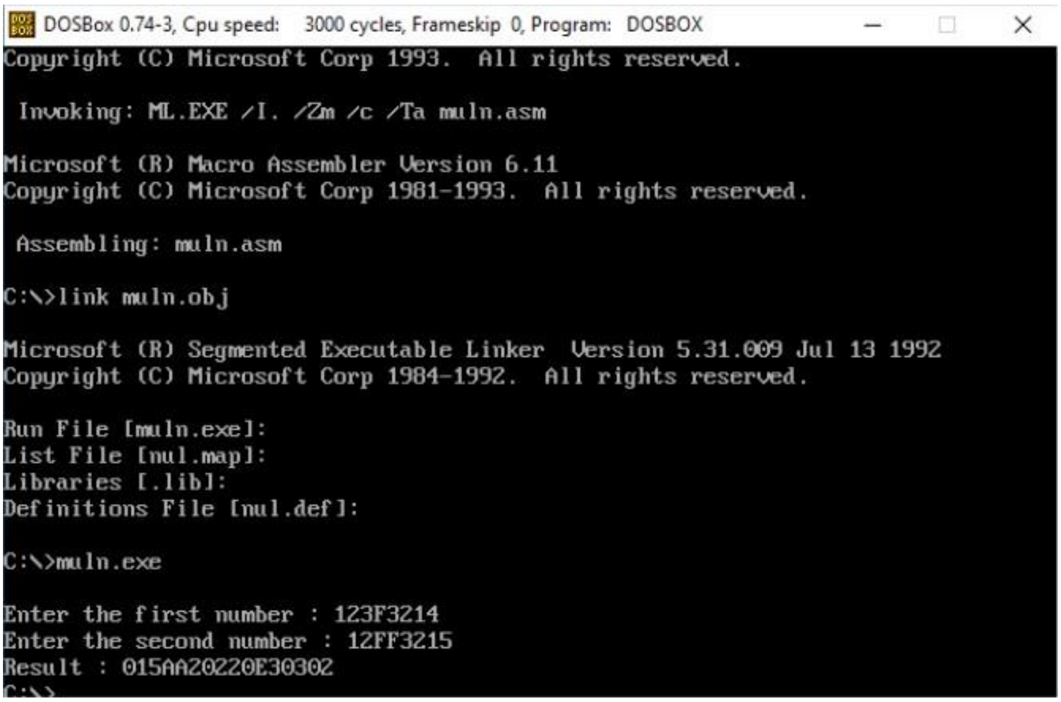
```
ab1:
and al,00fh
shl al,4
mov bl,al
;2nd  DIGIT OF FIRST NO.
mov ah,01
int 21h
cmp al,39h
jbe ab2
sub al,37h
ab2:
and al,00fh
add bl,al
loop abc
mov ad,ebx
;input of second 32 bit no.
mov dx,offset msg1
mov ah,09
int 21h
mov ebx,0
mov cx,4
abc1: shl ebx,8
;1st DIGIT OF FIRST NO.
mov ah,01
int 21h
cmp al,39h
jbe ab3
sub al,37h
```

```
ab3:

and al,00fh

shl al,4

mov bl,al

;2nd  DIGIT OF FIRST NO.

mov ah,01

int 21h

cmp al,39h

jbe ab4

sub al,37h

ab4:

and al,00fh

add bl,al

loop abc1

mov eax,ad

mul  ebx

mov ad,eax

mov ad1,edx

; printing

mov dx,offset msg2

mov ah,09

int 21h

mov ebx,ad1

mov cx,4

abc3 :rol ebx,8

mov al,bl

and al,0f0h

shr al,4
```

```
add al,30h
cmp al,39h
jbe ab5
add al,07h
ab5:
mov dl,al
mov ah,02
int 21h
mov al,bl
and al,00fh
add al,30h
cmp al,39h
jbe ab6
add al,07h
ab6:
mov dl,al
mov ah,02
int 21h
loop abc3
; printing
mov ebx,ad
mov cx,4
abc4 :rol ebx,8
mov al,bl
and al,0f0h
shr  al,4
add al,30h
cmp al,39h
```

```
jbe ab7

add al,07h

ab7:

mov dl,al

mov ah,02

int 21h

mov al,bl

and al,00fh

add al,30h

cmp al,39h

jbe ab8

add al,07h

ab8:

mov dl,al

mov ah,02

int 21h

loop abc4

.exit

End
```

OUTPUT:



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta muln.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: muln.asm

C:\>link muln.obj

Microsoft (R) Segmented Executable Linker  Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [muln.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

C:\>muln.exe

Enter the first number : 123F3214
Enter the second number : 12FF3215
Result : 015AA20220E30302
C:\>
```

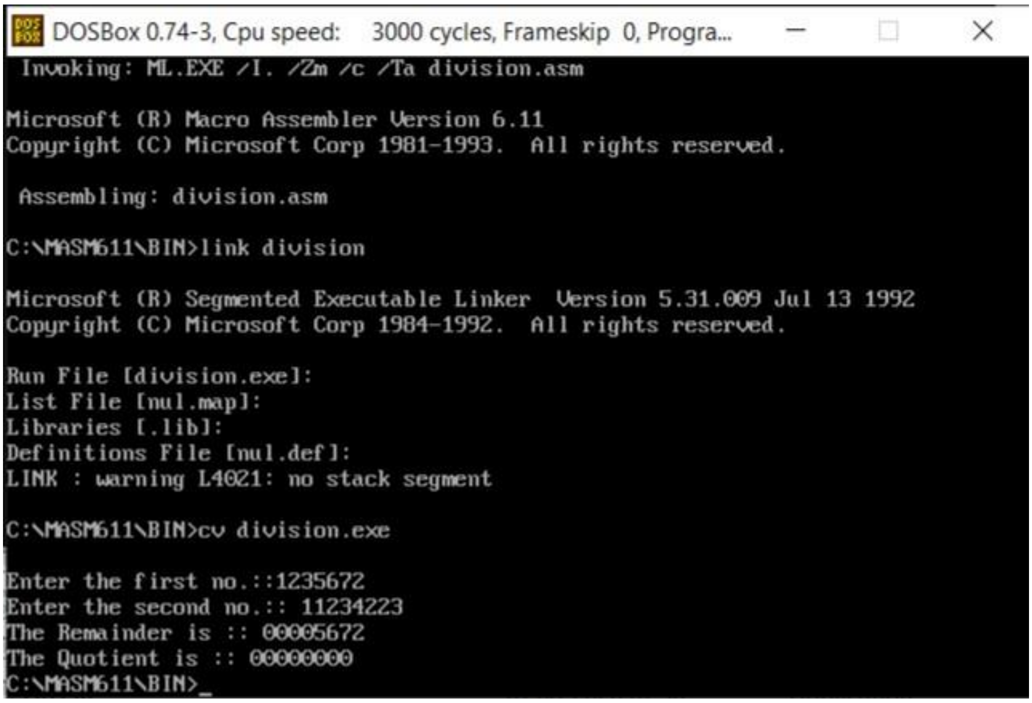
4. Division code:

```
.model small
.386
.data

DATA1 dd 00000000H
DATA2 dd 00000000H
REM dd ?
QUO dd ?
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Remainder is :: $"
msg3 db 10,13,"The Quotient is :: $"

.code
.startup
MOV AH,09
MOV DX,OFFSET msg
INT 21H
MOV EBX,0
MOV CX,8

AGAIN: MOV AH,01 ;1ST NO. ENTERED
INT 21H
CMP AL,'A'
JGE L5
JMP L6

L5: SUB AL,37H
L6: SUB AL,30H
SHL EBX,4
```

```
ADD  BL,AL
LOOP AGAIN
MOV DATA1,EBX
MOV AH,09
MOV DX,OFFSET msg1
INT 21H
MOV EBX,0
MOV CX,8
AGAIN1:MOV AH,01 ;2nd NO. ENTERED
INT 21H
CMP AL,'A'
JGE  L7
SUB AL,30H
JMP L8

L7: SUB AL,37H
L8: SHL EBX,4
ADD  BL,AL
LOOP AGAIN1
MOV DATA2,EBX
MOV EBX,0
MOV EDX,0
MOV EAX,0
MOV EAX,DATA1
MOV EBX,DATA2

DIV EBX
MOV REM,EDX ;REM=REMAINDER
MOV QUO,EAX ;QUO=QUOTIENT
MOV AH,09
MOV DX,OFFSET msg2
INT 21H
MOV EBX,REM
MOV CX,8

AGAIN2: ROL EBX,4
MOV DL,BL
AND DL,0FH ; to o/p the result in rem
CMP DL,9
JBE L1
ADD DL,37H
MOV AH,02
INT  21H
JMP L2

L1: ADD DL,30H
MOV AH,02
INT 21H

L2: LOOP AGAIN2
MOV AH,09
MOV DX,OFFSET msg3
INT 21H
MOV EBX,QUO
MOV CX,8

AGAIN3: ROL EBX,4
MOV DL,BL
AND DL,0FH ; to o/p the result in quo
CMP DL,9
JBE L3
ADD DL,37H
MOV AH,02
INT  21H
JMP L4

L3: ADD DL,30H
```

```
MOV AH,02
INT 21H

L4: LOOP AGAIN3
MOV AH,4CH
INT 21H
END
```

OUTPUT:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
 Invoking: ML.EXE /I. /Zm /c /Ta division.asm

Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: division.asm

C:\MASM611\BIN>link division

Microsoft (R) Segmented Executable Linker  Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [division.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
LINK : warning L4021: no stack segment

C:\MASM611\BIN>cv division.exe

Enter the first no.::1235672
Enter the second no.:: 11234223
The Remainder is :: 00005672
The Quotient is :: 00000000
C:\MASM611\BIN>_
```