Assignment 09 | MFP
CE-092

Assignment submission for Microprocessor Fundamentals and Programming subject week 9.

nevilparmar24@gmail.com

Task 1:

Write an assembly language to find the LCM of two numbers.

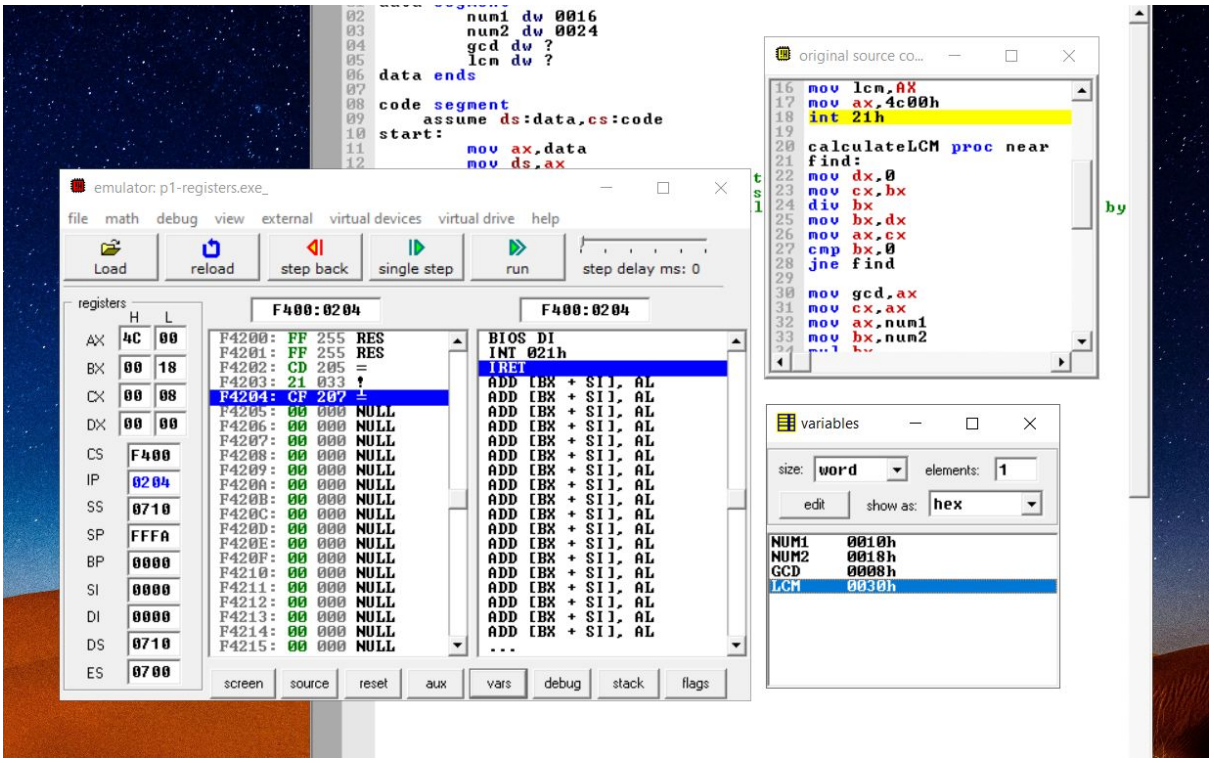
Passing params using registers.

Code:

```
data segment
        num1 dw 0016
        num2 dw 0024
        gcd dw ?
        lcm dw ?
data ends

code segment
    assume ds:data,cs:code
start:
        mov ax,data
        mov ds,ax
        mov ax,num1      ;NUM1 is stored in ax
        mov bx,num2      ; NUM2 is stored in bx
        call calculateLCM    ; calling the calculateLCM
procedure by passing the parameters in register
        mov lcm,AX
```

```
        mov ax,4c00h
        int 21h

calculateLCM proc near
find:
        mov dx,0
        mov cx,bx
        div bx
        mov bx,dx
        mov ax,cx
        cmp bx,0
        jne find

        mov gcd,ax
        mov cx,ax
        mov ax,num1
        mov bx,num2
        mul bx
        div cx
        RET
calculateLCM endp
code ends
end start
```

Output:



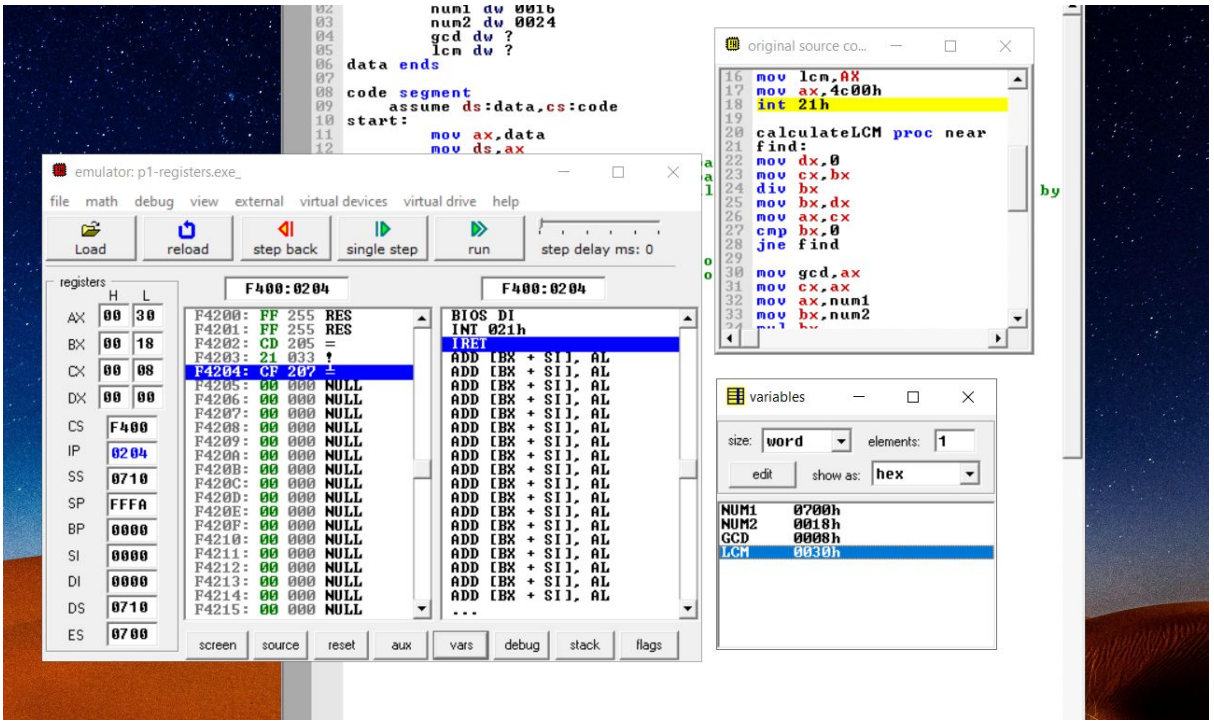
Passing params using pointers.

Code:

```
data segment
        num1 dw 0016
        num2 dw 0024
        gcd dw ?
        lcm dw ?
data ends

code segment
    assume ds:data,cs:code
start:
        mov ax,data
        mov ds,ax
        mov SI,offset num1      ; pass the offset of
```

```
num1 in SI
        mov DI,offset num2      ; pass the offset of
num2 in DI
        call calculateLCM     ; calling the calculateLCM
procedure by passing the parameters as a pointer
        mov ax,4c00h
        int 21h

calculateLCM proc near
        mov ax, [si] ; ax points to num1 in the memory
        mov bx, [di] ; bx points to num2 in the memory
find:
        mov dx,0
        mov cx,bx
        div bx
        mov bx,dx
        mov ax,cx
        cmp bx,0
        jne find

        mov gcd,ax
        mov cx,ax
        mov ax,[si]
        mov bx,[di]
        mul bx
        div cx
        mov lcm, ax
        RET
calculateLCM endp
code ends
end start
```

Output:



Passing params using stack.

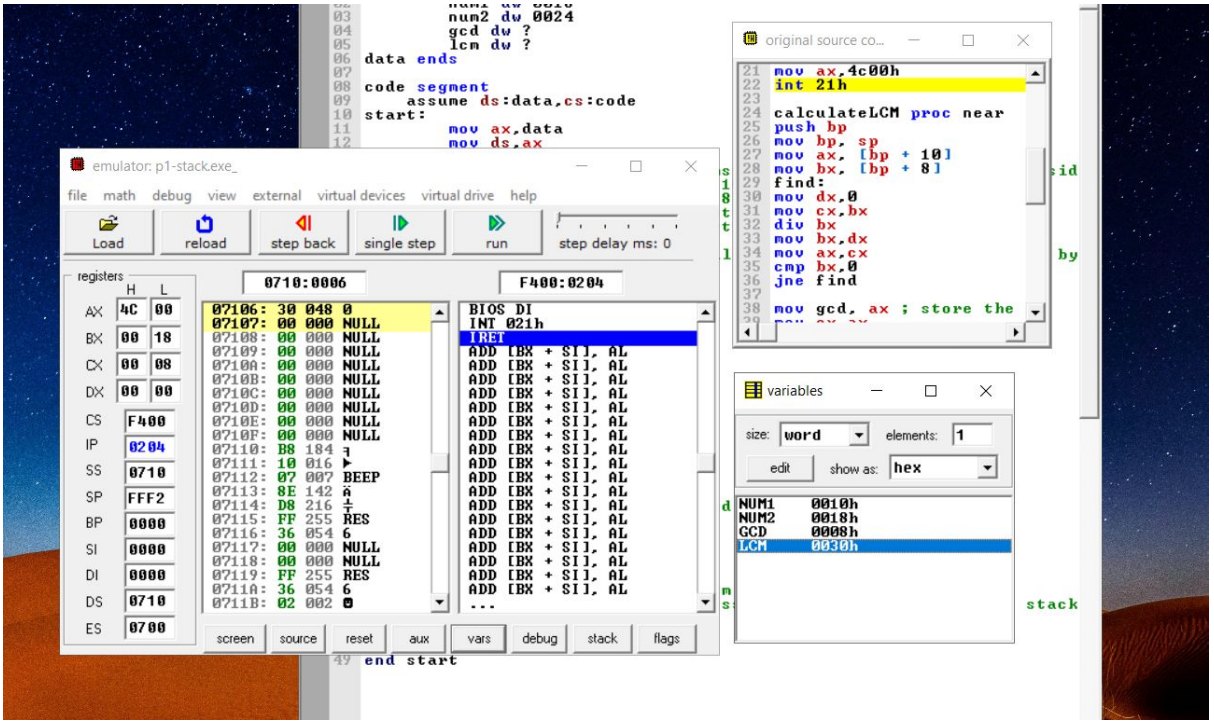
Code:

```
data segment
        num1 dw 0016
        num2 dw 0024
        gcd dw ?
        lcm dw ?
data ends

code segment
    assume ds:data,cs:code
start:
        mov ax,data
        mov ds,ax
```

```
        ; push the necessary params to the stack to
acces them inside the procedure instr
        push num1    ; offset SP + 10
        push num2    ; offset SP + 8
        push offset gcd     ; offset SP + 6
        push offset lcm     ; offset SP + 4

        call calculateLCM     ; calling the calculateLCM
procedure by passing the parameters as a pointer
        mov ax,4c00h
        int 21h

calculateLCM proc near
        push bp
        mov bp, sp
        mov ax, [bp + 10]
        mov bx, [bp + 8]
find:
        mov dx,0
        mov cx,bx
        div bx
        mov bx,dx
        mov ax,cx
        cmp bx,0
        jne find

        mov gcd, ax ; store the gcd
        mov cx,ax
        mov ax, [bp + 10]
        mov bx, [bp + 8]
        mul bx
        div cx
```

```
        mov lcm, ax ; store the lcm
        pop bp ; restore the address of BP from the top
of the stack
        RET
calculateLCM endp
code ends
end start
```

Output:



Task 2:

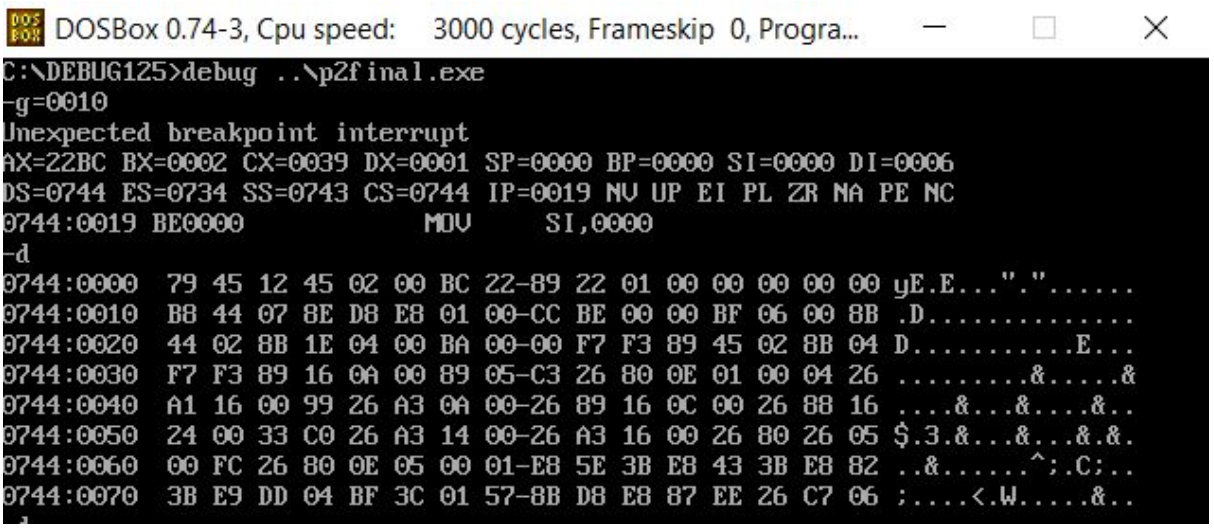
Write an assembly language to divide a 32 bit number by 16 bit number. The program should handle the quotient size exceeds 16 bits.

Code:

```
data segment
        num1 dd 45124579H
        num2 dw 0002h
        quotient dd ?
        remainder dw ?
data ends

code segment
    assume cs:code,ds:data
        mov ax,data
        mov ds,ax
        call division
        int 03

division proc near
        lea si,num1
        lea di,quotient
        mov ax,[si+2]
        mov bx,num2
        mov dx,0h
        div bx
        mov [di+2],ax
        mov ax,[si]
        div bx
        mov remainder,dx
        mov [di],ax
```

```
        ret
division endp
code ends
end
```

Output:



Nevil Parmar
CE-092
https://nevilparmar.me