Keegan Vaz
42
SECMPNB

EXPERIMENT 4

AIM: To write an assembly program to find the GCD of two numbers

ALGORITHM:

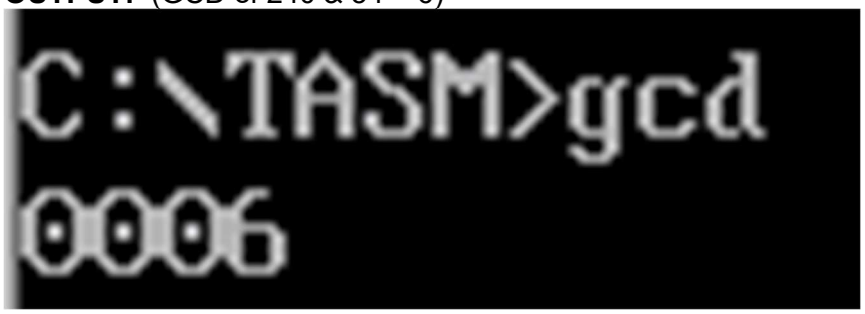
Step I : Initialize the data segment.
Step II : Load AX and BX registers with the operands.
Step III : Check if the two numbers are equal. If yes goto step X, else goto step IV.
Step IV : Is number 1 > number 2 ? If yes goto step VI else goto step V.
Step V : Exchange the contents of AX and BX register, such that AX contains the bigger number.
Step VI : Initialize DX register with 00H.
Step VII : Perform the division operation (contents of AX / contents of BX).
Step VIII : Check if there is remainder. If yes goto step IX, else goto step X.
Step IX : Move the remainder into AX register and goto step IV.
Step X : Save the contents of BX as GCD.
Step XI : Display the result.
Step XII : Stop.

CODE:

```
.model small
 .stack 100
 .data
 no1  dw 0240
 no2  dw 0054
 gcd dw 0h
 .code
     mov     ax,@data          ; initialize DS
     mov     ds, ax
     mov     ax, no1           ; get the first number
     mov     bx, no2           ; get the second number
again:
     cmp     ax, bx            ; check if nos are equal
     je      endd              ; if equal, save the GCD
     jb      exchg             ; if no,
                               ; is AX                    ; if yes interchange
 l2:
     mov     dx, 0
     div     bx                ; check if ax is
                               ; divisible by bx
     cmp     dx, 0     ;
     je      endd
     mov     ax, dx            ; mov the remainder
                               ; as no1 data
     jmp     again
 exchg :
     xchg     ax, bx
     jmp l2
 endd :
     mov     gcd, bx
     mov     ch, 04h           ; Count of digits to be
             ; displayed
     mov     cl, 04h           ; Count to roll by 4 bits
```

```
l12:
    rol     bx, cl        ; roll bl so that msb
                          ; comes to lsb
    mov     dl, bl        ; load dl with data
            ; to be displayed
    and     dl, 0fH       ; get only lsb
    cmp     dl, 09        ; check if digit is 0-9
                          ; or letter A-F
    jbe     l4
    add     dl, 07        ; if letter add 37H else
                          ; only add 30H
l4:
    add     dl, 30H
    mov     ah, 02        ; INT 21H
                          ; (Display character)
    int     21H
    dec     ch            ; Decrement Count
    jnz     l12
    mov     ah, 4ch
    int     21h
end
```

OUTPUT: (GCD of 240 & 54 = 6)



Keegan Vaz
42
SECMPNB

AIM: To write an assembly program to find the LCM of two numbers

ALGORITHM:

1. Start
2. Store first number(num1) in a register
3. Store second number(num2) in another register
4. Initialize a counter register(Rd) to 01h
5. Compare both the values num1 and num2
   - If num1 = num2 : Store num1 or num2 as result and jump to step 8
   - If num1 < num2 : Swap the register values so that num1 > num2
6. Multiply num2 and Rd and divide the product with num1
7. Check the reminder
   - If reminder is zero then store product obtained from multiplication in step 6 as result and jump to step 8
   - Else increment Rd and repeat steps 6 and 7
8. Stop

CODE:

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

read macro n,j1,j2
    mov cx,0ah
j1:mov ah,01h
    int 21h
    cmp al,0dh
    je j2
    sub al,30h
    mov bl,al
    mov ax,n
    mul cx
    xor bh,bh
    add ax,bx
    mov n,ax
    jmp j1
j2 :nop
endm

.model small
  .stack 100h

  .data
    msg1 db 10,13,'Enter the 1st number: $'
    msg2 db 10,13,'Enter the 2nd number: $'
    msg3 db 10,13,'The LCM= $'
    data1 dw 0
    data2 dw 0
```

```
    dat1 dw 0
    dat2 dw 0

.code
main proc

    mov ax,@data
    mov ds,ax
    print msg1

    ;reading 1st multidigit number
    read data1,jump1,jump2

    print msg2
    ;reading 2nd multidigit number
    read data2,jump3,jump4

    ;copy the data1 and data2 to dat1& dat2
    mov bx,data1
    mov dat1,bx

    mov cx,data2
    mov dat2,cx

    ;Algorithm for finding lcm
    ;if(dat1=dat2) then  finish, lcm=dat1 or dat2
    ;elseif(dat1<dat2) then  dat1=dat1+data1
    ;else dat2=dat2+data2
    ;repeat
loop1:mov ax,dat1
    cmp ax,dat2
    je jump5
    jc jump6

    mov ax,dat2
    add ax,cx
    mov dat2,ax
    jmp loop1

jump6:mov ax,dat1
    add ax,bx
    mov dat1,ax
    jmp loop1

    ;printing LCM
jump5:mov bx,0ah
    xor cx,cx

    ;push into stack
p1:xor dx,dx
    div bx
    push dx
    inc cx
    cmp ax,00h
    jne p1
```

```
    print msg3
    ;pop from stack
display:pop dx
    add dl,30h
    mov ah,02h
    int 21h
    loop display
    mov ah,4ch
    int 21h

main endp

end
```

OUTPUT: