Alexandria University
Faculty of Engineering
Electrical Engineering Department
3rd year communications
Microprocessor-software
07/01/2021

Task 2

Emu8086

Name: Mahmoud Fawzy Taha-Elaraby

Sec: 6

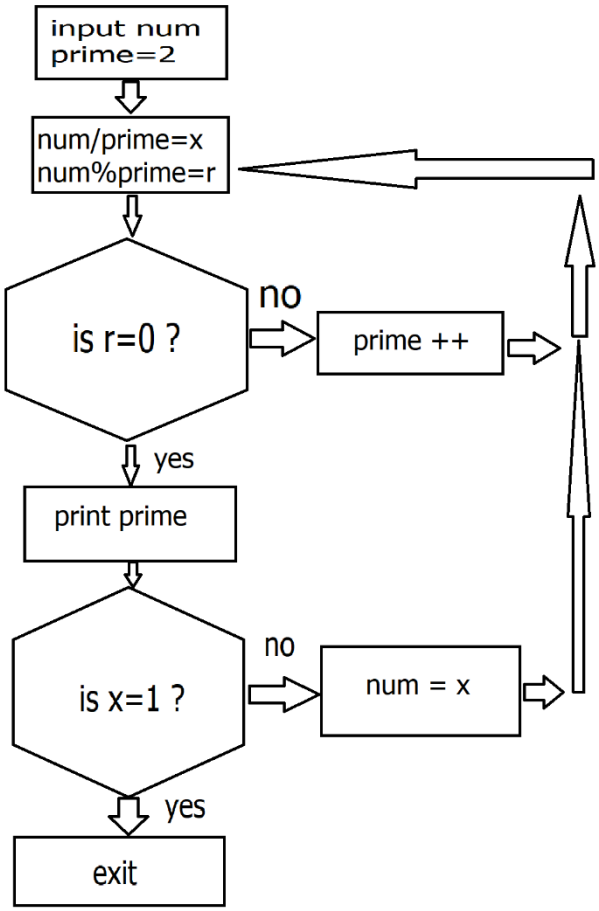
ID: 202

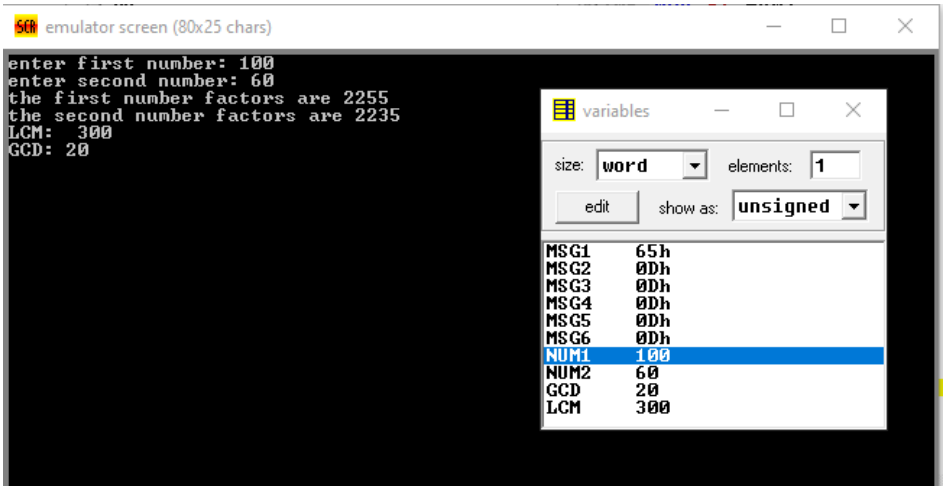
Part 1: Computing lcm and gcd:

```
include emu8086.inc
jmp start

msg1 db  'enter first number: $'
msg2 db  0dh,0ah , 'enter second number: $'
msg3 db  0dh,0ah , 'the first number factors are $'
msg4 db  0dh,0ah , 'the second number factors are $'
msg5 db  0dh,0ah , 'LCM:  $'
msg6 db  0dh,0ah , 'GCD: $'

num1 dw ?
num2 dw ?
GCD DW ?
LCM DW ?

start:

; taking first number from user and store it in num1
mov dx, offset msg1
mov ah, 9
int 21h

call    scan_num
mov     num1, cx

; taking second number from user and store it in num2
mov dx, offset msg2
mov ah, 9
int 21h

call    scan_num
mov     num2, cx

FACTOR1:
; print factors 1:

mov dx, offset msg3
mov ah, 9
int 21h

MOV AX,num1

MOV CL,AL
MOV BL,02H ; THE FACTOR IS 2
```

```
STEP1:
DIV BL  ; THE MAIN DIVIDED NUM IS IN AL & REMINDER IN AH
CMP AH,00H  ; IS AH = 0 ? (THE FACTOR IS CORRECT ?)
JE PRINT1    ; IF YES : GO AND PRINT IT
INC BL
MOV AL,CL
MOV AH,00H
JMP STEP1

PRINT1:
MOV CL,AL
MOV AL,BL
CALL   print_num
MOV AL,CL
CMP CL,01H  ; IS THE NUMBER = 1 NOW
JNE STEP1    ; IF NO, GO AND CONTINUE.  IF YES, THIS IS THE END

FACTOR2:
; print factors 2:

mov dx, offset msg4
mov ah, 9
int 21h

MOV AX,num2

MOV CL,AL
MOV BL,02H ; THE FACTOR IS 2

STEP2:
DIV BL  ; THE MAIN DIVIDED NUM IS IN AL & REMINDER IN AH
CMP AH,00H  ; IS AH = 0 ? (THE FACTOR IS CORRECT ?)
JE PRINT2    ; IF YES : GO AND PRINT IT
INC BL
MOV AL,CL
MOV AH,00H
JMP STEP2

PRINT2:
MOV CL,AL
MOV AL,BL
CALL   print_num
MOV AL,CL
CMP CL,01H  ; IS THE NUMBER = 1 NOW
JNE STEP2    ; IF NO, GO AND CONTINUE.  IF YES, THIS IS THE END
```

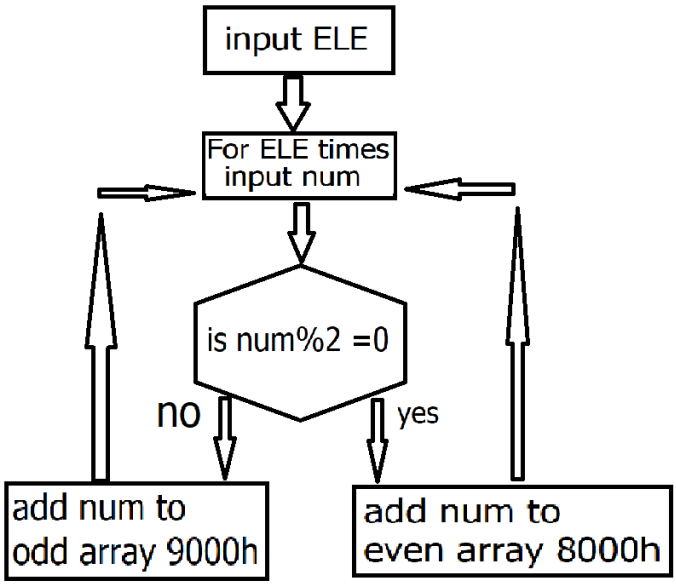
```
; GENERATING LCM & GCD
MOV AX,NUM1
MOV BX,NUM2

WHILE:
MOV DX,0
MOV CX,BX
DIV BX
MOV BX,DX
MOV AX,CX
CMP BX,0
JNE WHILE

MOV GCD,AX
MOV CX,AX

MOV AX,NUM1
MOV BX,NUM2
MUL BX
DIV CX
MOV LCM,AX

REST:
; print LCM
mov dx, offset msg5
mov ah, 9
int 21h
mov     ax, LCM
call    print_num

; print GCD:
mov dx, offset msg6
mov ah, 9
int 21h
mov     ax, GCD
call    print_num

HLT         ; halt!

DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS
```





Part 2: Sorting

```
include emu8086.inc
jmp start
msg1 db  'ENTER NUMBER OF ELEMENTS: $'
msg2 db  0dh,0ah , 'ENTER NUMBER: $'
msg3 db  0dh,0ah , 'NUMBER OF EVEN NUMS: $'
msg4 db  0dh,0ah , 'EVEN NUMS ARE:  $'
msg5 db  0dh,0ah , 'NUMBER OF ODD NUMS:  $'
msg6 db  0dh,0ah , 'ODD  NUMS ARE:  $'
ELE dw ?
num dw ?
EE dw 00H
EO dw ?
EVN DW 8000H
ODD DW 9000H

EVEN:  ;if num is even store it in 8000h and after
MOV BX,EVN
MOV AX,NUM
MOV [BX],AX
INC BX
MOV EVN,BX
INC DH  ; TO KNOW HOW MANY EVEN NUMBERS
PUSH DX
MOV DL,00H
MOV EE,DX
POP DX
JMP CONT

ODDD:  ;if num is odd  store it in 9000h and after
MOV BX,ODD
MOV AX,NUM
MOV [BX],AX
INC BX
MOV ODD,BX
JMP CONT

start:
; taking the total number of elements
mov dx, offset msg1
mov ah, 9
int 21h

call    scan_num
mov     ELE, cx

MOV BX,0001H
```




```
MOV DH,00H
MOV CH,00H

LOOP1:  ;separating odd and even
PUSH DX
MOV DH,00H
mov dX, offset msg2
mov ah, 9
int 21h
call    scan_num
mov     NUM, CX
POP DX
PUSH BX

MOV AX,NUM
MOV AH,00H
MOV BL,02H
DIV BL
CMP AH,00H ; IS REMINDER =0 (EVEN)
JE EVEN   ;if num is even store it in 8000h and after
JNE ODDD  ;if num is odd  store it in 9000h and after

CONT:
POP BX
CMP BX,ELE
JE SORT
INC BX
JMP LOOP1

SORT:
MOV AX,ELE
MOV BX,EE
SUB AL,BH
MOV AH,00H
MOV EO,AX
MOV BL,BH
MOV BH,00H
MOV EE,BX

PE:   ;to print  even
mov dx, offset msg3
mov ah, 9
int 21h
MOV AX,EE
call    print_num

mov dx, offset msg4
mov ah, 9
```

```
int 21h
MOV BX,8000H

PRINT1:
MOV AH,00H
MOV AL,[BX]
CMP AL,00H
JE PO
call    print_num
INC BX
JMP PRINT1

PO:    ;to print  odd
mov dx, offset msg5
mov ah, 9
int 21h
MOV AX,EO
call    print_num

mov dx, offset msg6
mov ah, 9
int 21h
MOV BX,9000H

PRINT2:
MOV AH,00H
MOV AL,[BX]
CMP AL,00H
JE END
call    print_num
INC BX
JMP PRINT2

END:
HLT         ; halt!
DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS
```