**Lab 06: (Open Lab # 1)**

**Perform the following tasks by designing and implementing the algorithms in Assembly Language. Make sure to attach the code (properly commented) and the screenshot of console (output), in your report. The report/document to be submitted should be renamed as *StudentName\_QalamID\_OpenLab\_A*.**

**TASK 1:**

**Write a program to check whether the input number is positive, negative or zero.**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

.model small

.data

msg1 db 10 ,13, "Enter the number$"

msg2 db 10,13, "Number is positive$"

msg3 db 10 ,13 , "Number is Negative$"

msg4 db 10, 13 , "Number is Zero$"

.code

main proc

mov ax,@data

mov ds,ax

mov dx,offset msg1 ;Prompt

mov ah,9

int 21h

mov ah, 1 ;input

int 21h

mov bl ,al

mov cl, 30h ; ascii or <or>

cmp bl, cl

jg positive

jl negative

je zero

positive:

lea dx,msg2

mov ah,9

int 21h

mov ah ,4ch

int 21h

negative:

lea dx,msg3

mov ah,9

int 21h

mov ah ,4ch

int 21h

Zero:

lea dx,msg4

mov ah,9

int 21h

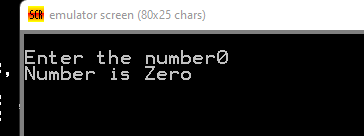
mov ah ,4ch

int 21h

main endp

ret

Output:

****

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**TASK 2:**

**Write a program to count the number of vowels in any given string.**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

.MODEL small

.STACK

.DATA

msg db 'Name :$'

result1 db 10,13, 'total char: $'

result db 10,13, 'counting vowels: $'

.CODE

.STARTUP

mov ax ,@data

mov ds,ax

mov ah,9

lea dx,msg

int 21h

mov bl,0

mov cl, 0

mov ah,1

int 21h

while:

cmp al ,0dh

je endl

cmp al, 'a'

je ok

cmp al, 'e'

je ok

cmp al, 'i'

je ok

cmp al, 'o'

je ok

cmp al, 'u'

je ok

up:

inc bl

int 21h

jmp while

ok:

inc cl

jmp up

endl:

mov ah,9

lea dx, result1

int 21h

mov ah ,2

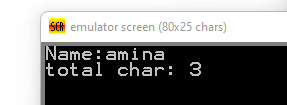
mov dl,cl

add dl,30h

int 21h

endp

ret



**TASK 3:**

**Write a program to find the ‘greatest common divisor’ (GCD) or ‘highest common factor’ (HCF) of two unsigned integers**

Program should load two registers with two Numbers  and then apply the logic for GCD of two Numbers . GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisor of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

mov bx,1100h ;set bx as pointer

mov DI,1200h ;set di register as pointer

mov ax,[bx]

mov cx,[bx+2]

rpt:

cmp ax,cx ;cpmare two data inputs

je store ;if data are equal store cx

jnc skip

xchg ax,cx

skip:

mov dx,0000h

div cx

cmp dx,0000h

je store

mov ax,dx

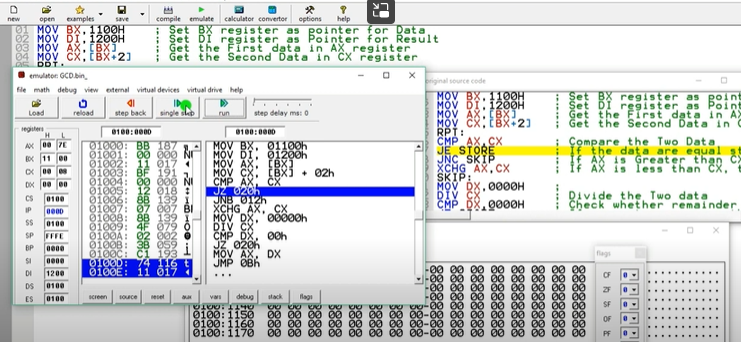
jmp rpt

store:

mov [dI],cx

hlt

ret



**TASK 4:**

**Write a program to swap two numbers in assembly language**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

.model small

.data

.code

main proc

mov ax,@data

mov ds,ax

mov ax,'1'

mov bx ,'2'

push ax

push bx

pop ax

pop bx

mov dx,ax

mov ah,2

int 21h

mov dx,bx

mov ah,2

int 21h

mov ah ,4ch

int 21h

main endp

ret

output:

