**Lab # 9**

**Tasks**

**Name:Laiba Gul**

**SapID:55740**

**Class:BSCS(7th)**

1. Write an assembly language that compares the values A and B and prints a message if the value in A is greater than B, less than B.

**Code:**

.model small

.stack 100h

.data

msg\_greater db 'A is greater than B$'

msg\_less db 'A is less than B$'

.code

main:

mov ax, @data

mov ds, ax

mov al, 7

mov bl, 5

cmp al, bl

jg A\_is\_greater

jl A\_is\_less

jmp exit

A\_is\_greater:

mov dx, offset msg\_greater

call print

jmp exit

A\_is\_less:

mov dx, offset msg\_less

call print

jmp exit

print:

mov ah, 09h

int 21h

ret

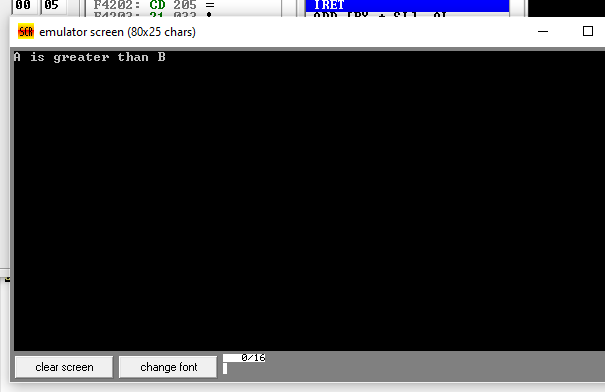
exit:

mov ah, 4Ch

int 21h

end main

**Output:**



1. Write an assembly language program that allow user to input one-digit number and determine the number is positive or zero.

**Code:**

.model small

.stack 100h

.data

msg\_zero db 'The number is ZERO$'

msg\_positive db 'The number is POSITIVE$'

.code

main:

mov ax, @data

mov ds, ax

mov ah, 01h

int 21h

sub al, '0'

cmp al, 0

je is\_zero

jg is\_positive

is\_zero:

mov dx, offset msg\_zero

call print

jmp exit

is\_positive:

mov dx, offset msg\_positive

call print

jmp exit

print:

mov ah, 09h

int 21h

ret

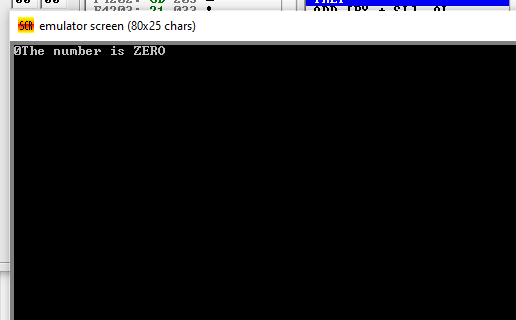
exit:

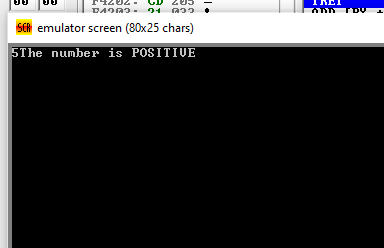
mov ah, 4Ch

int 21h

end main

**Output:**





1. Write an assembly language program that allow user to input one-digit number and determine if it is even or odd.

**Code:**

.model small

.stack 100h

.data

msg\_even db 'The number is EVEN$'

msg\_odd db 'The number is ODD$'

.code

main:

mov ax, @data

mov ds, ax

mov ah, 01h

int 21h

sub al, '0'

mov ah, 00h

mov bl, 2

div bl

cmp ah, 0

je is\_even

jne is\_odd

is\_even:

mov dx, offset msg\_even

call print

jmp exit

is\_odd:

mov dx, offset msg\_odd

call print

jmp exit

print:

mov ah, 09h

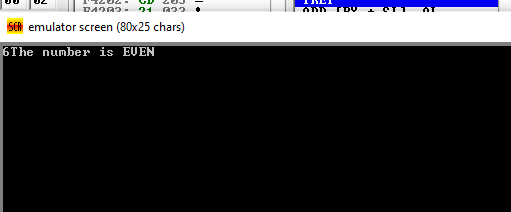
int 21h

ret

exit:

mov ah, 4Ch

int 21h

end main

1. Create an assembly program which takes your Marks as input, if your marks are less than 5 messages should be displayed “Need hard work” else print “Satisfactory”.

**Code:**

.model small

.stack 100h

.data

msg\_hardwork db 'Need hard work$'

msg\_satisfactory db 'Satisfactory$'

.code

main:

mov ax, @data

mov ds, ax

mov ah, 01h

int 21h

sub al, '0'

cmp al, 5

jl need\_hard\_work

jge satisfactory

need\_hard\_work:

mov dx, offset msg\_hardwork

call print

jmp exit

satisfactory:

mov dx, offset msg\_satisfactory

call print

jmp exit

print:

mov ah, 09h

int 21h

ret

exit:

mov ah, 4Ch

int 21h

end main

