



Lab Task 3

Course Title: Microprocessor and Microcontroller Lab

Course Code: CSE3102

Section: 3

Submitted To

Shakib Mahmud Dipto

Lecturer, Dept of CSE

University of Liberal Arts Bangladesh

Submitted By:

MD. Jahidul Haque Junayer [221014071]

Spring, 24

Problem 1 : Write a program to find the largest between two numbers.

Source Code

```
include 'emu8086.inc'

.model small
.stack 100h

.data
    a db ?

.code

newLine proc
    mov ah, 2
    mov dl, 10
    int 21h
    mov dl, 13
    int 21h
    ret

input proc
    mov ah, 1
    int 21h
    sub al, 48
    ret

output proc
    mov ah, 2
    add a, 48
    mov dl, a
    int 21h
```

```
ret
```

```
main proc
```

```
    mov ax,@data
```

```
    mov ds,ax
```

```
    print "Enter first number: "
```

```
    call input
```

```
    mov a, al
```

```
    call newLine
```

```
    print "Enter second number: "
```

```
    call input
```

```
    cmp a,al
```

```
    JG Abiggest
```

```
    JL Bbiggest
```

```
Bbiggest:
```

```
    mov a, al
```

```
    call newLine
```

```
    call output
```

```
    print " is the bigger"
```

```
    jmp exit
```

```
Abiggest:
```

```
    call newLine
```

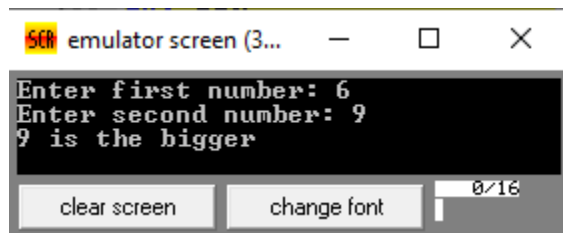
```
    call output
```

```
    print " is the bigger"
```

```
exit:
    mov ah,4ch
    int 21h

main endp
end main
```

Output



Problem 2 : Write a program to find the smallest between two numbers.

Source Code

```
include 'emu8086.inc'
.model small
.stack 100h

.data
    a db ?

.code

newLine proc
    mov ah, 2
    mov dl, 10
    int 21h
```

```
mov dl, 13
```

```
int 21h
```

```
ret
```

```
input proc
```

```
mov ah, 1
```

```
int 21h
```

```
sub al, 48
```

```
ret
```

```
output proc
```

```
mov ah, 2
```

```
add a, 48
```

```
mov dl, a
```

```
int 21h
```

```
ret
```

```
main proc
```

```
mov ax, @data
```

```
mov ds, ax
```

```
print "Enter first number: "
```

```
call input
```

```
mov a, al
```

```
call newLine
```

```
print "Enter second number: "
```

```
call input
```

```
cmp a, al
```

JL Asmallest

JG Bsmallest

Bsmallest:

mov a, al

call newLine

call output

print " is the smaller"

jmp exit

Asmallest:

call newLine

call output

print " is the smaller"

exit:

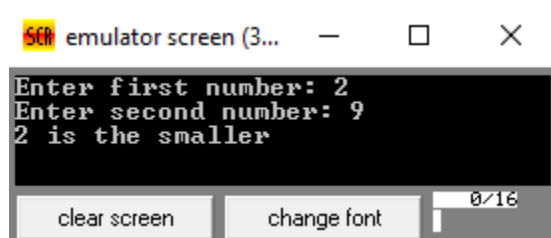
mov ah,4ch

int 21h

main endp

end main

Output



The screenshot shows a window titled "emulator screen (3...". The window contains a black terminal area with white text. The text reads: "Enter first number: 2", "Enter second number: 9", and "2 is the smaller". Below the terminal area, there are two buttons: "clear screen" and "change font". To the right of these buttons is a small text box containing "0/16".

```
Enter first number: 2
Enter second number: 9
2 is the smaller
```

clear screen change font 0/16

Problem 3 : Write a program which character comes first in alphabetic order between two characters.

Source Code

```
include 'emu8086.inc'
.model small
.stack 100h

.data
    char1 db ?
    char2 db ?
    msg_char1_first db 'First character comes first.', '$'
    msg_char2_first db 'Second character comes first.', '$'
    msg_same_char db 'Characters are the same.', '$'

.code

newLine proc
    mov ah, 2
    mov dl, 10
    int 21h
    mov dl, 13
    int 21h
    ret

main proc
    mov ax, @data
    mov ds, ax

    print "Enter first character: "; Input first character
```

```
mov ah, 1
```

```
int 21h
```

```
mov char1, al
```

```
call newLine
```

```
print "Enter second character: "; Input second character
```

```
mov ah, 1
```

```
int 21h
```

```
mov char2, al
```

```
call newLine
```

```
; Compare characters
```

```
mov al, char1
```

```
mov bl, char2
```

```
;call newLine
```

```
cmp al, bl
```

```
jb char1_first
```

```
je same_char
```

```
jmp char2_first
```

```
char1_first:
```

```
mov ah, 9
```

```
lea dx, msg_char1_first
```

```
int 21h
```

```
jmp exit_prog
```

```
char2_first:
```

```
mov ah, 9
```

```
lea dx, msg_char2_first
```



```
int 21h
jmp exit_prog

same_char:
    mov ah, 9
    lea dx, msg_same_char
    int 21h
    jmp exit_prog

exit_prog:
    mov ah, 4Ch
    int 21h
main endp

end main
```

Output

