# Name: Asad Haroon Reg #: SP17-BCS-012

# Lab 10 & 11 – Using Basic Instructions: A Programming Exercise II

## Objectives

In this lab, students are assigned a programming exercise related to what they have learned in previous labs. Also, viva will be conducted from each student.

## Lab Tasks

**Task 1:** Write a program to (a) display a “?”, (b) read two decimal digits whose sum is less than 10, (c) display them and their sum on the next line, with an appropriate message.

*Sample Execution:*

?27

THE SUM OF 2 AND 7 is 9.

**Task 2:** Write a program to (a) prompt the user, (b) read first, middle, and last initials of a person’s name, and (c) display them down the left margin.

*Sample Execution:*

Enter three initials: JFK

J

K

F

**Task 3:** Write a program to read one of the hex digits A-F, and display it on the next line in decimal.

*Sample Execution:*

Enter a hex digit: C

In decimal it is 12

**Task 4:** Write a program to display a 10\*10 solid box of asterisk. ***Hint:*** Declare a string in the data segment that specifies the box, and display it with INT 21h, function 9h?

***Task-1 Solution:***

org 100h

.MODEL SMALL

.DATA

MSG db 'The SUM of$'

msg2 db ' And$'

msg3 db ' is$'

.CODE

MAIN PROC

mov ax,@DATA

mov ds,ax

mov ah,2

mov dl,'?' ;printing ? on screen first

int 21h

mov ah,1

int 21h ; input first number

mov bl,al

mov ah,1

int 21h ;input second number

mov bh,al

mov cl,bh

add cl,bl ;store the sum

mov ah,2

mov dl,0Ah ;next line

int 21h

mov ah,2

mov dl,0Dh ; carriage return

int 21h

lea dx,MSG ; display msg of 'The sum of'

mov ah,9

int 21h ; display till $

mov ah,2

mov dl,bl ;printing first number

int 21h

lea dx,msg2 ;'AND' msg

mov ah,9

int 21h

mov ah,2

mov dl,bh ;printing second number

int 21h

lea dx,msg3 ; 'IS' MSG

mov ah,9

int 21h

mov ah,2

sub cl,48 ;printing sum of two numbers

mov dl,cl

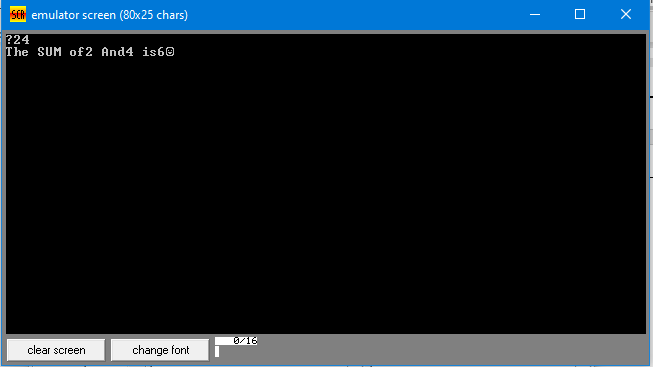
int 21h

mov ah,2

mov dl,01h

int 21h

ret



***Task-2 Solution:***

org 100h

.MODEL SMALL

.DATA

MSG db 'Enter three initials$'

.CODE

MAIN PROC

mov ax,@DATA

mov ds,ax

lea dx,MSG

mov ah,9

int 21h

mov ah,1

int 21h

mov bl,al

mov ah,1

int 21h

mov bh,al

mov ah,1

int 21h

mov cl,al

mov ah,2

mov dl,0Ah

int 21h

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,bl

int 21h ;first letter

mov ah,2

mov dl,0Ah

int 21h ; next line after first letter

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,bh

int 21h ;second letter

mov ah,2

mov dl,0Ah

int 21h ; next line after second letter

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,cl

int 21h ;third letter

mov ah,2

mov dl,0Ah

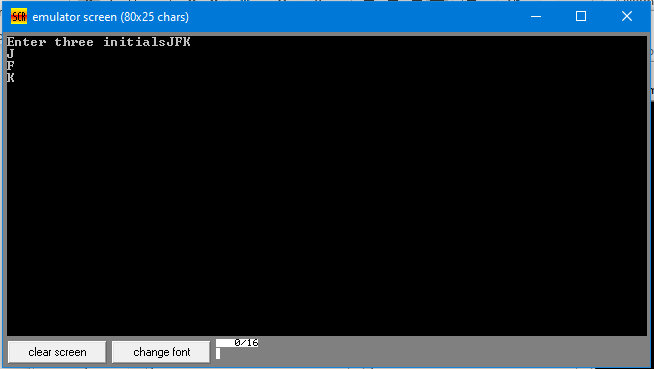
int 21h ; next line after third letter

mov ah,2

mov dl,0Dh

int 21h

ret



***Task-3 Solution:***

org 100h

.MODEL SMALL

.STACK 100h

.DATA

MSG db 'Enter a hex digit:$'

output db 'IN decimal it is:1$'

exitL db 'You have not entered Char in range A-F$'

.CODE

mov ax,@DATA

mov ds,ax

lea dx,MSG

mov ah,9

int 21h

mov ah,1

int 21h

mov bl,al

sub bl,11h

cmp al,40h ;checking Input char is in Caps or not

jle exit

cmp al,47h ;checking Input char is in Caps or not

jge exit

mov ah,2

mov dl,0Dh

int 21h

mov ah,2 ;spacing

mov dl,0Ah

int 21h

lea dx,output ;printing output msg

mov ah,9

int 21h

mov ah,2

mov dl,bl

int 21h

mov ah,4ch

int 21h

exit:

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,0Ah

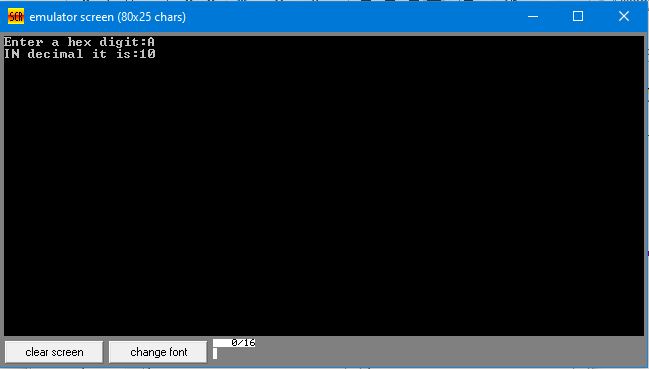
int 21h

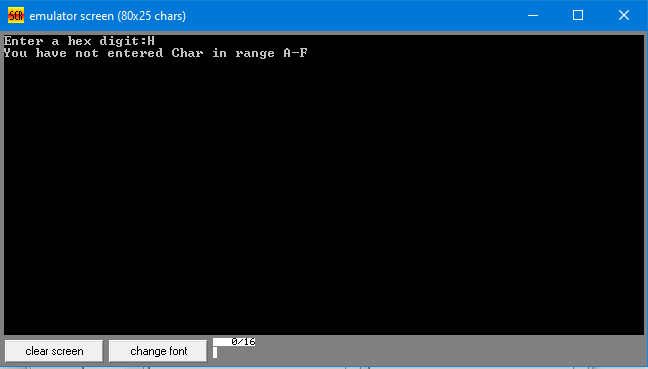
lea dx,exitL ;inputted char is not in A-F

mov ah,9

int 21h

ret





***Task-4 Solution:***

org 100h

.MODEL SMALL

.STACK 100h

.DATA

rows db '\* \*$'

row1 db '\*\*\*\*\*\*\*\*\*\*$'

.CODE

mov ax,@DATA

mov ds,ax

lea dx,row1

mov ah,9

int 21h

mov cl,8

L1:

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,0Ah

int 21h

lea dx,rows

mov ah,9

int 21h

loop L1

mov ah,2

mov dl,0Dh

int 21h

mov ah,2

mov dl,0Ah

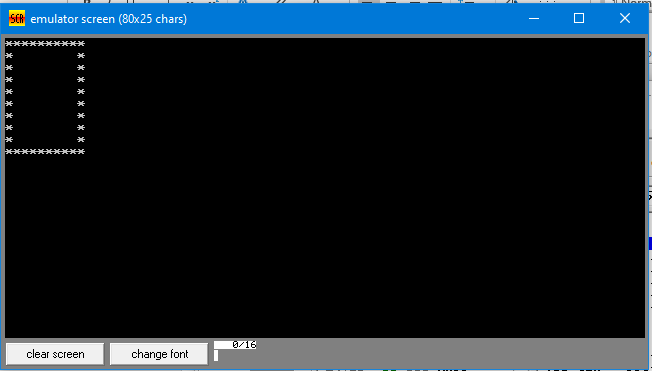
int 21h

lea dx,row1

mov ah,9

int 21h

ret



# Lab 12 – Processor Status and Flag Register

## Objectives

The main objective of this lab is to observe the effects on processor status flag bits after execution of different assembly language instructions.

## Lab Tasks

For each of the following program, write down the contents of only those registers and flag bits which have been modified after execution of each instruction. Do, state the reason as well when flags bits modified. Assume flag bits are unset at start of each program.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Instructions** | **Register Contents** | **Flags bits and reasons** |
|  | MOV AX, 0020h | 0020 | IF=1, its default value if interrupt will call then IF will become 0. |
| MOV BX, 00AAH | 00AA | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| MOV AH, BL | 00 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
|  | MOV AL,81 | 51 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are zero. |
| ADD AL,0FEh | FE | IF=1, its default value if interrupt will call then IF will become 0.  CF=1.Because if we add 81 and 0FEh then it will generate the carry at end so CF=1.  And other flags are 0. |
|  | MOV AX, 5510h | 5510 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are zero. |
| SUB AL,2 | 0E | IF=1, its default value if interrupt will call then IF will become 0.  AF=1, because it take some borrow or generate carry then its value is 1 in this case 5510h-2h is taking borrow 16 at end to make 16h-2h=14h=E so, its answer is 0E.  And other flags are 0. |
|  | MOV AL,0FEh | FE | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| SUB AL,2 | FC | SF=1, because it generate signed number and F is signed negative so it generate SF=1.  PF=1,And total number of 1’s in AL is even so PF=1.  IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| MOV BL,8Ch | 8C | IF=1, its default value if interrupt will call then IF will become 0.  SF=1, because it generate signed number and 8 is signed negative so it generate SF=1.  PF=1, total number of 1’s in BL is even so PF=1.  And other flags are 0. |
| MOV BH,2Dh | 2D | IF=1, its default value if interrupt will call then IF will become 0.  SF=1, because it generate signed number and 2 is signed negative so it generate SF=1.  PF=1, total number of 1’s in BL is even so PF=1.  And other flags are 0. |
| ADD BX, AX | 2E88 | PF=1, total number of 1’s in BL is even so PF=1.  IF=1, its default value if interrupt will call then IF will become 0.  AF=1, because it take some borrow or generate carry then its value is 1. In this case, 2D8C+00FC, when C+C made then they generate 1 carry.  And other flags are 0. |
|  | MOV AX, 1234 | 04D2 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| MOV BX, AX | 04D2 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| MOV CX, AX | 04D2 | IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| ADD CH, AL | D6 | SF=1, because in D6, D is signed negative.  IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| ADD AX, 0FFFFh | 04D1 | CF=1, because in addition carry generated.  PF=1.because in lower byte AL, number of 1’s are even.  AF=1, because it take some borrow or carry to make sum.  IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |
| DEC BX | 04D1 | CF=1, because in addition carry generated.  PF=1.because in lower byte BL, number of 1’s are even.  IF=1, its default value if interrupt will call then IF will become 0. |
| NEG CX | 292E | CF=1, because in taking 2’s complement carry generated.  PF=1.because in lower byte CL, number of 1’s are even.  AF=1, because it take some borrow or carry to make sum.  IF=1, its default value if interrupt will call then IF will become 0.  And other flags are 0. |