

*Lab Report*

Assembly Language Lab

*Submitted By*

Mobassir Al Masud Basunia

ID: 151-15-4899

Section: D

Date: 11.12.16

Question One: Get input from user and print it.

*Code:*

org 100h

.model small

.stack 100h

.code

mov ah, 01h

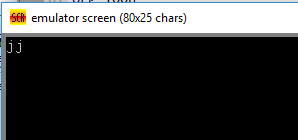
int 21h

mov dl, al

mov ah, 02h

int 21h

*Output:*



*Question Two:* Addition, subtraction in assembly language, solve the equations in assembly.

1. A + A – B + C
2. 2 \*A - B

*Code:*

org 100h

.data

NUM1 DB ?

NUM2 DB ?

NUM3 DB ?

NUM4 DB ?

NUM5 DB ?

NUM6 DB ?

RESULT DB ?

MSG DB "ENTER 1 FOR ADDTION 2 FOR SUBTRACTION ", 0AH, 0DH, "3(a+a-b+c) AND 4(2a-b) FOR MULTIPLE OPARATION, ", 0AH, 0DH, "CHOOSE: $"

MSG1 DB "ENTER NUMBERS : $"

MSG2 DB " + $"

MSG4 DB " - $"

MSG5 DB " \* $"

MSG3 db "RESULT: $"

SH2 DB "2 \* $"

ENDS

.code

LEA DX,MSG

MOV AH,9

INT 21H

MOV AH, 01H

INT 21H

CMP AL, 31H

JZ Addition

CMP AL, 32H

JZ Subtraction

CMP AL, 33H

JZ Multiple

CMP AL, 34H

JZ Multiplication

Addition:

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG1

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM1,AL

LEA DX,MSG2

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM2,AL

ADD AL,NUM1

MOV RESULT,AL

MOV AH,0

AAA

ADD AH,30H

ADD AL,30H

MOV BX,AX

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG3

MOV AH,9

INT 21H

MOV AH,2

MOV DL,BH

INT 21H

MOV AH,2

MOV DL,BL

INT 21H

MOV AH,4CH

INT 21H

Subtraction:

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG1

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM1,AL

LEA DX,MSG4

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM2,AL

MOV AL,NUM1

SUB AL,NUM2

MOV RESULT,AL

MOV AH,0

AAA

ADD AH,30H

ADD AL,30H

MOV BX,AX

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG3

MOV AH,9

INT 21H

MOV AH,2

MOV DL,BH

INT 21H

MOV AH,2

MOV DL,BL

INT 21H

MOV AH,4CH

INT 21H

Multiple:

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG1

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV BL,AL

LEA DX,MSG2

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM2,AL

LEA DX,MSG4

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM3,AL

LEA DX,MSG2

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM4,AL

ADD BL, NUM2

SUB BL,NUM3

ADD BL, NUM4

MOV AL, BL

MOV RESULT,AL

MOV AH,0

AAA

ADD AH,30H

ADD AL,30H

MOV BX,AX

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX,MSG3

MOV AH,9

INT 21H

MOV AH,2

MOV DL,BH

INT 21H

MOV AH,2

MOV DL,BL

INT 21H

MOV AH,4CH

INT 21H

Multiplication:

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX, MSG1

MOV AH, 09H

INT 21H

LEA DX,SH2

MOV AH,9

INT 21H

MOV AH, 01H

INT 21H

MOV NUM1, AL

LEA DX,MSG4

MOV AH,9

INT 21H

MOV AH, 01H

INT 21H

MOV NUM2, AL

MOV BL, 2H

MOV AL, NUM1

MUL BL

MOV BL, NUM2

SUB AL, BL

MOV NUM3, AL

MOV AH, 02H

MOV DL, 0DH

INT 21H

MOV DL, 0AH

INT 21H

LEA DX, MSG3

MOV AH, 09H

INT 21H

MOV DL, NUM3

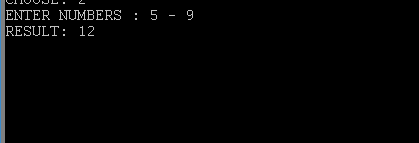
MOV AH, 02H

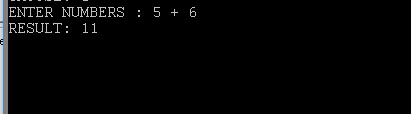
INT 21H

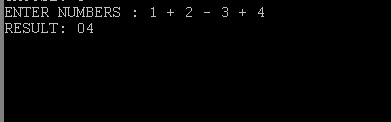
RET

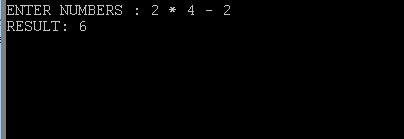
ret

*Output:*









*Question Three:* Implement an assembly code to

* + take 2 single character input
  + Store them in 2 variables
  + Convert the uppercase one into Lowercase
  + Convert the Lowercase one into uppercase
  + Display them

*Code:*

org 100h

.model small

.data

str1 db 'Enter Lower Case: $'

str2 db 'Enter Upper Case: $'

str3 db 'Result: Lower Case to Upper Case: $'

str4 db ' Upper Case to Lower Case: $'

strx db ?

stry db ?

.code

main proc

lea dx, str1

mov ah, 09h

int 21

mov ah, 01h

int 21h

sub al, 20h

mov strx, al

; new line

mov ah, 02h

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

lea dx, str2

mov ah, 09h

int 21h

mov ah, 01h

int 21h

add al, 20h

mov stry, al

; new line

mov ah, 02h

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

lea dx, str3

mov ah, 09h

int 21h

mov dl, strx

mov ah, 02h

int 21h

lea dx, str4

mov ah, 09h

int 21h

mov dl, stry

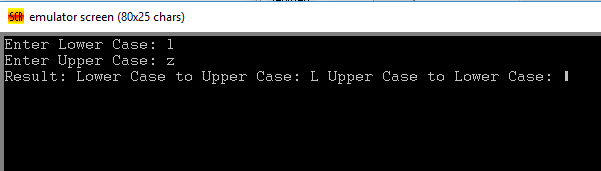
mov ah, 02h

int 21h

endp

end main

*Output:*



*Question Four:* Take an integer input from keyboard and print corresponding square of stars.

*Code:*

; square

org 100h

.model small

.stack 100h

.data

str1 db "\*$"

str2 db " $"

str3 db 0ah, 0dh, "$"

var db 1

va db 0

strenter db "Enter a number between 1-9: $"

.code

main proc

lea dx, strenter

mov ah, 09h

int 21h

mov ah, 01h

int 21h

sub al, 48

mov va, al

mov ah, 09h

lea dx, str3

int 21h

mov cl, va

add va, 1

mov bl, va

l1:

l2:

mov dl, str1

mov ah, 02h

int 21h

mov dl, str2

mov ah, 02h

int 21h

loop l2

mov ah, 09h

lea dx, str3

int 21h

inc var

mov al, va

cmp var, al

je ter

mov cx, bx

loop l1

ter:

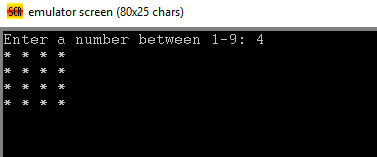
mov ah, 4ch

int 21h

endp

end main

*Output:*



*Question Five:* Take an integer input from keyboard and print corresponding triangle of stars.

*Code:*

; square

org 100h

.model small

.stack 100h

.data

str1 db "\*$"

str2 db " $"

str3 db 0ah, 0dh, "$"

var db 1

va db 0

strenter db "Enter a number between 1-9: $"

.code

main proc

lea dx, strenter

mov ah, 09h

int 21h

mov ah, 01h

int 21h

sub al, 48

mov va, al

mov ah, 09h

lea dx, str3

int 21h

mov cl, va

add va, 1

mov bl, va

l1:

l2:

mov dl, str1

mov ah, 02h

int 21h

mov dl, str2

mov ah, 02h

int 21h

loop l2

mov ah, 09h

lea dx, str3

int 21h

dec bx

mov cx, bx

cmp cx, 0

jz ter

loop l1

ter:

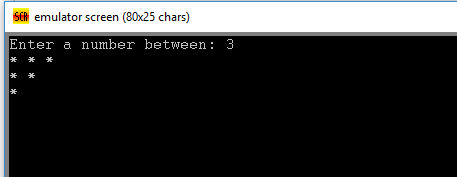
mov ah, 4ch

int 21h

endp

end main

*Output:*



*Question Six:* Take an integer input from keyboard and print corresponding triangle of stars.

*Code:*

.CODE

main proc

MOV CX,3

MOV BX,1

loop1:

MOV CX, BX

loop2:

CMP CX, 0

JZ done

mov ah, 2

mov dl, '\*'

int 21h

loop loop2

inc bx

cmp bx, 4

je done

mov ah, 02h

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

loop loop1

ENDP

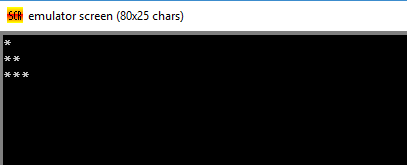
done:

MOV AH,4CH

INT 21H

RET

*Output:*



*Question Seven:* Take an integer input from keyboard and print corresponding diamond of stars.

*Code:*

org 100h

.stack 100h

.data

i db ?

j db ?

k db ?

num db '1'

C1 db 1

input db ?

C2 db ?

.code

.startup

mov ah, 1

int 21h

mov j, al

mov input, al

mov ah, 2

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

L1:

mov bl, j

mov i, bl

cmp i, '1'

je L2

space\_loop:

mov dl, ' '

mov ah, 2

int 21h

dec i

cmp i, '1'

jne space\_loop

L2:

mov bl, C1

mov k, bl

NL:

mov al, '\*'

mov bl, 2

div al

cmp al, 0

je ps

mov dl,'\*'

mov ah, 2

int 21h

ps:

mov dl, ' '

mov ah, 2

int 21h

dec k

cmp k, 0

jg NL

add C1, 1

inc num

mov ah, 2

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

dec j

cmp j, '0'

jne L1

mov bl, input

mov j, bl

mov bl, 1

mov C1, bl

dec num

mov bl, input

sub bl, 30h

mov C2, bl

L3:

mov bl, C1

mov i, bl

space\_loop2:

mov dl, ' '

mov ah, 2

int 21h

dec i

cmp i, 0

jne space\_loop2

inc C1

mov bl,C2

mov k, bl

dec num

L4:

mov al, '\*'

mov bl, 2

div al

cmp al, 0

je L5

mov ah, 2

mov dl, '\*'

int 21h

L5:

mov ah, 2

mov dl, ' '

int 21h

dec k

cmp k, 1

jne L4

dec C2

mov ah, 2

mov dl, 0dh

int 21h

mov dl, 0ah

int 21h

dec j

cmp j, '1'

jne L3

mov ah, 2

mov dl, 0dh

int 21h

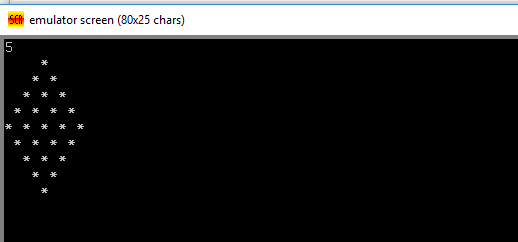
mov dl, 0ah

int 21h

.exit

ret

*Output:*



*Question Eight:* Take a character input from keyboard and print corresponding binary value and count the number of 1’s in that binary value (using loop and rotation operation)

*Code:*

.model small

.stack 100h

.code

main proc

mov bh,0

mov ah,1

int 21h

mov bl,al

mov cx,8

top:

rol bl,1

jc e

jnc e1

e:

mov ah,2

mov dl,49

int 21h

inc bh

jmp e2

e1:

mov ah,2

mov dl,48

int 21h

e2:

loop top

mov ah,2

mov dl,0ah

int 21h

mov dl,0dh

int 21h

add bh,48

mov ah,2

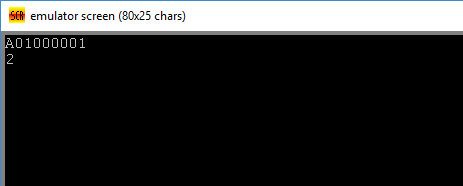
mov dl,bh

int 21h

main endp

end main

*Output:*



*Question Nine:* Write an assembly program to reverse a number of 10 digits. Also display an error message if the user is giving input anything except numbers (0-9).

*Code:*

org 100h

include emu8086.inc

org 100h

.model small

.stack 100h

.data

data1 db 0

v dw 0

cc db 0

tstcase db "Enter test case: $"

pri db 0ah, 0dh, "Enter number: $"

outx db 0ah, 0dh, "Output: $"

.code

lea dx, tstcase

mov ah, 09h

int 21h

call scan\_num

mov data1, cl

lea dx, pri

mov ah, 09h

int 21h

mov cl, data1

l1:

mov ah, 01h

int 21h

mov bx, ax

cmp bl, 0dh

jz termin

mov dl, 10

mov al, 30h

check:

cmp bl, al

je ok

inc al

dec dl

cmp dl, 0

jnz check

jnz nook

nook:

print "Error!"

cmp cl, 0

jz output

loop l1

ok:

push bx

cmp cl, 0

jz output

loop l1

output:

lea dx, outx

mov ah, 09h

int 21h

mov cl, data1

l2:

pop bx

mov ah, 02h

mov dl, bl

int 21h

dec cx

cmp cx, 0

jnz l2

termin:

mov ah, 4ch

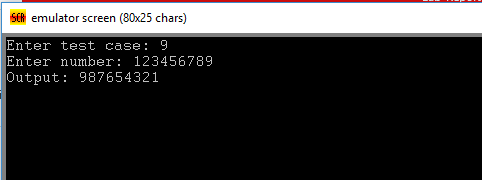
int 21h

DEFINE\_SCAN\_NUM

end

ret

*Output:*



*Question Ten:* Write an assembly program to take a multi digit number (4 digit number) and display it.

*Code:*

include emu8086.inc

org 100h

.model small

.stack 100h

.data

data1 db 10

data2 dw 0

inx db "Enter number (Enter to continue): $"

outx db 0ah, 0dh, "Output: $"

.code

lea dx, inx

mov ah, 09h

int 21h

mov cx, 0

mov bl, 00h

l1:

mov ah, 1

int 21h

cmp al, 0dh

jz ter

mov ah, 0

mov data2, ax

sub data2, 30h

mov dx, data2

mov data2, bx

mov ax, data2

mul data1

mov bx, ax

add bx, dx

loop l1

ter:

lea dx, outx

mov ah, 09h

int 21h

mov ax, bx

call print\_num

ret

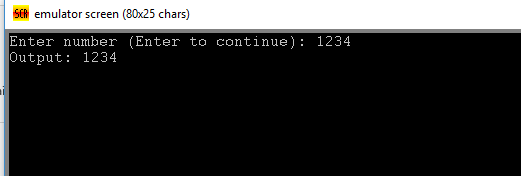
DEFINE\_SCAN\_NUM

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

end

*Output:*



*Question Eleven:* Take inputs from keyboard

* Print “Digit “ if it is Numeric value
* Print “Uppercase “if it is Uppercase Letter
* Print “Lowercase” if it is Lowercase Letter

*Code:*

org 100h

.model small

.stack 100h

.data

vari db 0

digit db "Digit$"

ucase db "Uppercase$"

lcase db "Lowercase$"

error db "Input is a not a digit nor uppercase nither lowercase.$"

str1 db "Enter something: $"

str2 db 0ah, 0dh, "Result: $"

.code

main proc

lea dx, str1

mov ah, 09h

int 21h

mov ah, 01

int 21h

mov vari, al

mov cl, 10

mov bl, 30h

l1:

cmp al, bl

jz show1

inc bl

loop l1

mov cl, 26

mov bl, 41h

l2:

cmp al, bl

jz show2

inc bl

loop l2

mov cl, 26

mov bl, 61h

l3:

cmp al, bl

jz show3

inc bl

loop l3

jnz show4

show1:

mov vari, al

lea dx, str2

mov ah, 09h

int 21h

lea dx, digit

mov ah, 09h

int 21h

ret

show2:

mov vari, al

lea dx, str2

mov ah, 09h

int 21h

lea dx, ucase

mov ah, 09h

int 21h

ret

show3:

mov vari, al

lea dx, str2

mov ah, 09h

int 21h

lea dx, lcase

mov ah, 09h

int 21h

ret

show4:

mov vari, al

lea dx, str2

mov ah, 09h

int 21h

lea dx, error

mov ah, 09h

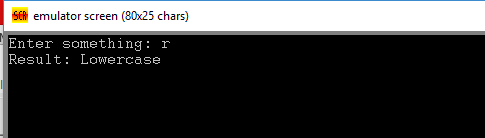
int 21h

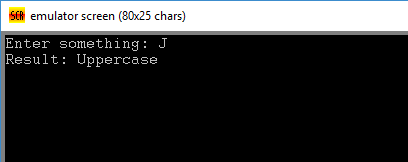
ret

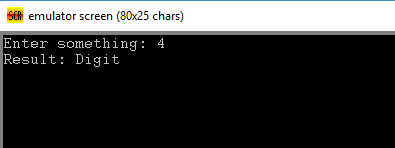
endp

end main

*Output:*







*Question Twelve:* Take an integer input from keyboard and print corresponding triangle of stars

*Code:*

org 100h

.model small

.stack 100h

.data

str db "Enter a number: $"

str1 db "\*$"

str2 db " $"

var db 1

va db ?

.code

lea dx, str

mov ah, 09h

int 21h

mov ah, 01h

int 21h

sub al, 48

mov va, al

mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

mov cl, 1

mov bl, 2

add va, 2

l1:

l2:

mov dl, str1

mov ah, 02h

int 21h

mov dl, str2

mov ah, 02h

int 21h

loop l2

inc bx

mov cx, bx

cmp bl, va

je ter

mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

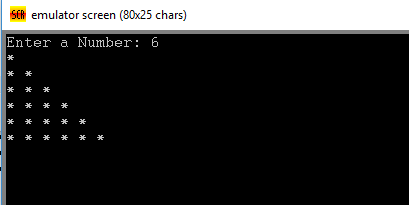
loop l1

ter:

mov ah, 4ch

int 21h

*Output:*



*THE END*