

① Write an Assembly Language Program of 8086 to display the message on the screen.

; Assembly Language Program to display welcome message

ASSUME CS:CODE, DS:DATA

DATA SEGMENT

; Beginning of Data Segment

MSG DB "WELCOME TO ASSEMBLY LANGUAGE PROGRAMMING"

DATA ENDS

CODE SEGMENT

START:

MOV AX, DATA

MOV DS, AX

; Initialise the data segment

LEA DX, MSG

; Offset Address of the msg into DX

MOV AH, 09H

; Load 09H into AH

INT 21H

; Invoke dos function call 09H

MOV AH, 4CH

INT 21H

; Control returns DOS Prompt

CODE ENDS

END START

19891A1225

- G. Bhuvan  
Sai  
Teja

## EXECUTION

C:\19891A~1 > cd 8086

C:\19891A~1 > 8086 > masm welcome.asm

Object filename <sup>welcome</sup> ~~[welcome.obj]~~ [welcome.obj]

Source listing [NUL.LST]:

Cross reference [NUL.CRF]:

C:\19891A~1\8086 > link welcome.~~obj~~.obj

Run File [welcome.exe]:

List File [NUL.MAP]:

Libraries [C.LIB]:

LINK : warning L4021: no stack segment.

C:\19891A~1\8086 > welcome.exe

WELCOME TO ASSEMBLY LANGUAGE PROGRAMMING

② Write an Assembly Language Program of 8086 for addition of two 8 bit numbers.

; assembly language program for addition

; ALP for adding two 8-bit numbers.

ASSUME CS:CODE DS:DATA

DATA SEGMENT

OPR1 DB 02H

OPR2 DB 03H

RESULT DB ?

DATA ENDS

; beginning of Data segment  
; operand 1

; operand 2.

; a byte of memory reserved  
for result

CODE SEGMENT

START:

; beginning of code segment

MOV AX, DATA

MOV DS, AX

; initiate data segment

MOV AL, OPR1

MOV BL, OPR2

CLC

ADD AL, BL

;  $AL \leftarrow AL + BL$

MOV DI, OFFSET RESULT

MOV [DI], AL

MOV AH, 4CH

INT 21H

CODE ENDS

END START

## Execution

C:\198914~1\8086 > masm add.asm

Object filename [add.OBJ]:

Source Listing [NUL.LST]:

Cross-reference [NUL.CRF]:

C:\198914~1\8086 > link add.obj

Run File [ADD.EXE]:

List File [NUL.MAP]:

Libraries [.LIB]:

C:\198914~1\8086 > debug add.exe.

-U

076B:0000 B86407 -MOV AX,0764

076B:000F BF0200 MOV DI,0002

-d 076A:0000 0002

076A:0000 02 03 00

-g

Program terminated normally

-d 076A:0000 0002

076A:0000 02 03 05

-q.



③ Write an Assembly language program of 8086 for subtraction of two 8-bit numbers

ASSUME CS:CODE DS:DATA

DATA SEGMENT

OPR1 DB 07H

OPR2 DB 04H

RESULT DB ?

DATA ENDS

CODE SEGMENT

START: MOV AX, DATA

MOV DS, AX

MOV AL, OPR1

MOV BL, OPR2

CLC

SUB AL, BL

MOV DI, OFFSET RESULT

MOV ~~CODE~~, AL

MOV AH, 4CH

INT 21H

CODE ENDS

END START

; beginning of data segment

; define variable OPR1

; define variable OPR2.

; define variable RESULT

; end of data segment

; beginning of code segment

; initialize data segment

; AL ← operand 1

; BL ← operand 2.

; clear carry flag

; AL ← AL - BL

; load address of result into DI

; store result in address in DI

; load dos func. into AH

; invoke dos function call

; end of code segment

; end of the program.

## Execution

C:\198914~1\8086 > <sup>asm</sup>masm sub.~~asm~~

Object Filename [sub.OBJ]:

Source Listing [NUL.LST]:

Cross-reference [NUL.CRF]:

C:\198914~1\8086 > ~~masm~~ link sub.obj

Run File [sub.exe]:

List File [NUL.MAP]:

Libraries [.LIB]:

C:\198914~1\8086 > debug sub.exe

-u

076B:0000 B86A07 MOV AX,076A.

076B:000F 8F0200 MOV DI,0002

-d 076A:0000 0002

076A:0000 07 04 00

-g

Program terminated ~~successfully~~ normally

-d 076A:0000 0002

076A:0000 07 04 03

-q

④ Write an assembly Language Program of 8086 for multiplication of two 8-bit numbers.

ASSUME CS:CODE DS:DATA

DATA SEGMENT

OPR1 DB 02H

OPR2 DB 08H

RESULT DB 01 DUP(?)

DATA ENDS

CODE SEGMENT

START: MOV AX, DATA

MOV DS, AX

MOV AL, OPR1

MOV BL, OPR2

MUL BL

MOV DI, OFFSET RESULT

MOV [DI], AL

MOV AH, 4CH

INT 21H

CODE ENDS

BND START

## Execution

C:\1989\14~1\8086> masm mul.asm

Object file name [mul.obj]:

Source listing [mul.lst]:

Cross-reference [mul.crf]:

C:\1989\14~1\8086> link mul.obj

Run File [mul.exe]:

List File [mul.map]:

Libraries [.lib].

C:\1989\14~1\8086> debug mul.exe.

-u

076A:0000 B86A07 MOV AX,076A

076B:000E BF0200 MOV DI,0002

-d

076A:0000 0002

076A:0000 02 03 00

-g

Program terminated normally.

-d 076A:0000 0002

0

076A:0000 02 03 06

-q.



8) Write an assembly language program for 8086 to ~~divide~~ divide 16 bit number by an 8-bit number.

ASSUME CS:CODE DS:DATA

DATA SEGMENT

A DW 0004H

B DB 02H

Q DB ?

R DB ?

DATA ENDS

CODE SEGMENT

START: MOV AX, DATA

MOV DS, AX

MOV AX, A

MOV BL, B

DIV BL

LEA DI, Q

MOV [DI], AL

LEA DI, R

MOV [DI], AH

MOV AH, 4CH

INT 21H

CODE ENDS

END START

; remainder in AH,  
quotient in AL.  
; get offset of Q.  
; load quotient into Q  
; get offset of R.

## Execution

C:\198914~1\8086 > masm div.asm

Object filename [div.obj]:

Source listing [NUL.lst]:

Cross-reference [NUL.crf]:

C:\198914~1\8086 > link div.obj

Run File [div.exe]:

List File [NUL.map]:

Libraries [ .LIB ]:

C:\198914~1\8086 > debug div.exe.

-u

076B:0000 886407 mov AX,076A

076B:0014 8D3E0400 lea DI,[0004]

-d. 076A:0000 0004

076A:0000 04 00 02 00 00

-g

Program terminated normally.

-d 076A:0000 0004

076A:0000 04 00 02 02 00

-q

⑥ Write an assembly language program for 8096 to find average of array elements.

ASSUME CS:CODE DS:DATA

DATA SEGMENT

ARRAY DB 1,4,2,3,8,6,7,5,9

AUG DB ?

MSG DB "AVERAGE = \$

DATA ENDS

CODE SEGMENT

START:

MOV AX, DATA

MOV DS, AX.

LEA SI, ARRAY

LEA DX, MSG

MOV AH, 09H

INT 21H

MOV AX, 00

MOV BL, 9

LOOP1:

ADD AL, ARRAY[SI]

INC SI

LOOP LOOP1

DI V BL

ADD AL, 30H

MOV DL, AL

MOV AH, 2

INT 21H

MOV AH, 4CH

INT 21CH

CODE ENDS

END START



## Execution

C:\19891A~1\8086 > masm avg.asm,

object filename [avg.obj]:

source listing [NUL.LST]:

Cross-reference [NUL.CRF]:

C:\19891A~1\8086 > link avg.<sup>obj</sup>.asm.

Run File [AVG.exe]:

List ~~File~~ File [NUL.MAP]:

Libraries [-LIB]:

C:\19891A~1\8086 > avg.exe.

AVERAGE = 5