

a.)

```

a. DATA           SEGMENT
                  0-3   4-7
DD      1111, 2222  4+4 = 8
                  8-9   10-11  12-13  14-15  16-17
DW      111, 222, 333, 444, 555  2+2+2+2+2 = 10
                  18   19   20   21   22   23   24
DB      11, 22, 33, 44, 55, 66, 77  1+1+1+1+1+1+1 = 7
                  31   32   33   34
DB      'MIDTERM', 13, 10, '$'  1+1+1+1 = 4
DATA          ENDS

```

DATA SEGMENT ข้างต้นใช้เนื้อที่รวมทั้งหมด.....35 bytes (0.5 คะแนน)

**DB [ Define byte ]** - 1 byte

**DW [ Define word ]** - 2 byte

**DD [ Define Doubleword ]** - 4 byte

b.)

ใช้ผล run โดย symbolic debugger (SYMDEB) ด้านล่างตอบคำถามข้อ b-i

ผล run โดย symbolic debugger (SYMDEB) เป็นดังนี้ คือ

-t

AX = 2FDC	BX = 026B	CX = 020C	DX = EA9B	SP = 0080
BP = 0012	SI = 0015	DI = 0027	DS = 0EAE	ES = 0E96
SS = 0EA6	CS = 0EAF	IP = 000F NV UP EI PL NZ NA PO NC		
0EAF:000F	F7F3	DIV BX		

b. คำสั่ง DIV BX เป็นคำสั่งขนาด.....2 bytes (0.25 คะแนน)

object code ของคำสั่ง DIV BX คือ.....F7F3 (0.25 คะแนน)

c. actual address ของคำสั่งที่จะถูก execute เป็นคำสั่งต่อไป ได้แก่ (ตอบเป็นเลขฐาน 16)

(1 คะแนน)

■ การคำนวณ actual address ที่ถูก  
 1. เนื่อง 0H ไม่ใช่ว่ามีช่อง segment address  
     ที่ 1CA20H  
 2. นำ offset .0008H  
 ให้ผลลัพธ์ คือ 1CA28H  
 ∴ actual address คือ 1CA2:0008 คือ 1CA28H

0EAFOH  
 000F H  
 0EAFFH \*

d. segment:offset ของคำสั่งที่อยู่ต่อจากคำสั่ง DIV BX คือ

(1 คะแนน)

$$0EAF : 000F \rightarrow \begin{array}{r} 000f \\ - 0011 \\ \hline 2 \end{array}$$

(by common sense จริงๆ)

คำสั่งต่อจาก DIV BX คือ segment : offset  
0EAF : 0011 #

in Lecture ที่พูดถึงห้องเรียน  
มาก มาก มาก )

e. ถ้า DH มีข้อมูลชนิด signed number ข้อมูลใน DH เป็นฐาน 10 คือ

(1 คะแนน)

$$DX = \underline{\underline{EA9B}}_{16}^{DH}$$

$$\begin{aligned} &= 1110\ 1010 \\ &\text{2's complement} \quad 0001\ 0101 \\ &\quad 1 \\ &= 0001\ 0110 \\ &= -22 \end{aligned}$$

f. ถ้า DL มีข้อมูลชนิด unsigned number ข้อมูลใน DL เป็นฐาน 10 คือ

(1 คะแนน)

1

$$DL = 1001\ 1011_2$$

จําลอง unsigned number 709 DL คือ  $1001\ 1011_2$

$$\therefore DL \text{ ฐาน } 10 = 155_{10}$$

g. operand [SI]+8 มี addressing mode เรียกว่า ..... Direct indexed addressing (0.5 คะแนน)

AX = 2FDC BX = 0268 CX = 020C DX = E998 SP = 0080  
BP = 0012 SI = 0015 DI = 0027 DS = 0EA6 ES = 0E96  
SS = 0EA6 CS = 0EAF IP = 0009 NV UP EI PL NZ NA PO NC  
0EAE0009H F7H3 DN = BX

0015H

29

16

13

h. actual address ของ operand [SI]+8 ได้แก่ (ตอบเป็นเลขฐาน 16)

$$\begin{aligned} 81+8 &= 29D \\ &= 1DH \end{aligned}$$

0EAEOH  
1DH  
0EAFDH \*

Yinuk  
segment address

i. actual address ของ operand [BP][DI] ได้แก่ (ตอบเป็นเลขฐาน 16)

$$\begin{aligned} 0039 &\text{ then } 0EA60H \\ &0039H \\ &0EA99H * \end{aligned}$$

j. NEW PROC

...

RET 10

NEW ENDP

10

procedure NEW มี distance attribute แบบ ..... (0.5 คะแนน)

คำสั่ง RET 10 จะ return ออกจาก procedure และ pop ..... words ออกจาก stack (0.25 คะแนน)

2. จงหาค่าสุดท้ายของ register หลังการ execute คำสั่งต่อไปนี้ โดยแสดงวิธีทำ (ถ้าไม่แสดงวิธีทำจะไม่ได้คะแนน)

(คำตอบละ 0.5 คะแนน)

a.	MOV	BX,0	$\text{BX} = 0$
	MOV	DI,8	$\text{DI} = 8$
	MOV	AX,573	$\text{AX} = 573$
MORE :	MOV	DX,0	$\text{DX} = 0$
	DIV	DI	$573/8 = ...$
	ADD	BX,DX	$0+0 = 0; \text{BX} = 0$
	CMP	AX,0	$... > 0$
	JNE	MORE	

BX มีข้อมูลคือ ..... 0 ..... (ตอบเป็นเลขฐาน 10)

DX มีข้อมูลคือ ..... 0 ..... (ตอบเป็นเลขฐาน 10)

## หน้าที่ 3

b. DATA

SEGMENT

ONE offset = 0 DB 'MIDTERM-EXAM'

TWO offset = 12 DB 'START-TODAY'

DATA ENDS

MOV SI,8 SI = 8

LEA BX,TWO BX = 12

ADD BX,1 BX = 13

MOV CX,4 CX = 4

NEXT: MOV DL,[SI] DL = 'E' , DL = 'X' DL = 'A' DL = 'M'

MOV [BX],DL [13] = 'E' , [14] = 'X' [15] = 'A' [16] = 'M'

INC SI SI = 9 , SI = 10 , SI = 11 , SI = 12 ,

INC BX BX = 14 , BX = 15 , BX = 16 , BX = 17 ,

LOOP NEXT CX = 3 , CX = 2 , CX = 1 , CX = 0 ,

MOV DI,2 DI = 2

MOV AL,TWO[DI] AL = TWO+2 ('A')

DL มีข้อมูลคือ ..... 'M' ..... (ตอบเป็น character)

AL มีข้อมูลคือ ..... 'X' ..... (ตอบเป็น character)

BX มีข้อมูลคือ ..... 17 ..... (ตอบเป็นเลขฐาน 10)

## c. DSEG SEGMENT

ONE DB 2,3,4,5,6,7,8 *7 byte*  
 TWO DW 11,12,13,14,15,16,17,18,19 *2x9 = 18 byte*

```
MOV SI,TWO SI=11
SUB SI,6 SI=5
MOV AL,[SI] AL=7
LEA BX,TWO BX=7 (Address of TWO:7)
MOV CX,[BX] CX=11
ADD CX,4 CX=15
ADD BX,CX BX=22
```

AL มีข้อมูลคือ..... *7* .....(ตอบเป็นเลขฐาน 10)

BX มีข้อมูลคือ..... *22* .....(ตอบเป็นเลขฐาน 10)

## 3. จงเปลี่ยนเป็นคำสั่งในภาษา Assembly

```
IF AL >= BL
  DI = DI+SI ADD DI,SI
ELSE
  DI = DI+BL ADD DI+BL
END IF ENDS
```

```
CMP AL,BL
JGE GREATER
JMP LESS
GREATER: ADD DI,SI
          JMP NEXT
LESS:   MOV AL,BL
          CBW
          MOV BX,Ax
          ADD DI,BX
NEXT:   ...
```

สมมุติว่ามีข้อมูลอยู่แล้วใน register AL,BL,DI,SI ซึ่งเป็นข้อมูลชนิด signed number

(3.5 คะแนน)

4. จงเขียนคำสั่งในภาษา Assembly เพื่อคำนวณ  $SUM = (BL/7)+(N*5)$  โดยให้ N และ SUM เป็น symbolic reference ใน data segment และ N มีขนาด 1 byte SUM มีขนาด 1 word สมมุติว่ามีข้อมูลอยู่แล้วใน register BL และที่ symbolic reference N ซึ่งเป็นข้อมูลชนิด signed number (ผลหารเอาเฉพาะส่วนที่เป็นจำนวนเต็มมาคำนวณ)

(4.5 คะแนน)

4. จงเขียนคำสั่งในภาษา Assembly เพื่อคำนวณ  $SUM = (BL/7)+(N*5)$  โดยให้ **N** และ **SUM** เป็น symbolic reference ใน data segment และ **N** มีขนาด 1 byte **SUM** มีขนาด 1 word สมมุติว่ามีข้อมูลอยู่แล้วใน register BL และที่ symbolic reference N ซึ่งเป็นข้อมูลชนิด signed number (ผลหารเอาเฉพาะส่วนที่เป็นจำนวนเต็มมาคำนวณ)

(4.5 คะแนน)

```
MOV AL, BL ; AL=BL  
MOV CL, 7 ; CL=7  
IDIV CL ; AX=AL/7 (AH:低字节 AL:高字节)  
CBW ; แปลง Ax=AH:低字节 AL:高字节 (High byte)  
MOV BX, AX ; หันไปที่ BX  
MOV AL, N ; AL=N  
MOV CL, 5 ; CL=5  
IMUL CL ; AX= AL×CL  
ADD BX, AX ; BX= BX+AX  
MOV SUM, BX ; SUM= BX
```