EASTERN MEDITERRANEAN UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 323: Microprocessors

Midterm Exam

Lecturer: Prof.Dr. Hasan Kömürcügil

Date: 23 / 11 / 2016

Time Allowed: 100 minutes

Name and Surname: SOLUTIONS

Student Number: ....

- There are 5 questions in this exam paper.
- Answer all questions.
- Write clearly and tidily.
- Correct answers without sufficient explanation might not get full points!
- Mobile phones must be switched off in the exam room.

Question	Points Gained
Q1 *(17 points)*	
Q2 *(20 points)*	
Q3 *(23 points)*	
Q4 *(20 points)*	
Q5 *(20 points)*	
Total	

Q1) [17 *points*]

a) ***[11 points]*** Consider the following data segment:

```
            ORG 004AH
DATA1       DB          'EMU'
DATA2       DD          286A9B7H
DATA3       DW          10, 0C35FH
```

Show the offset address and the contents of the corresponding memory locations.

Memory Offset Add. (in Hex)	Memory Contents (in Hex)
004A	'E'
004B	'M'
004C	'U'
004D	B7
004E	A9
004F	86
0050	02
0051	0A
0052	00
0053	5F
0054	C3

b) ***[6 points]*** If DS=9027H and BX=4DE2H, calculate the physical address (PA) of the code segment.

PA= 90270 + 4DE2 = 95052H

Q2) [20 *points*]

Trace the following program segment and find the contents of AL, DATA2, CX and CF.

```
        .DATA
DATA1 DB 32H, 4AH, 8BH
DATA2 DB 3 DUP (?)
        .CODE
        MOV CX, 3
        MOV BX, Offset DATA1
        MOV DI, Offset DATA2
NEXT:   MOV AL, [BX]
        MOV AH, AL
        AND AL, 81H
        JZ K1
        CMP AL, 81H
        JNZ K2
K1:     MOV DL, [BX]
        MOV [DI], DL
        INC DI
K2:     INC BX
        LOOP NEXT
```

AL= 81H (4)

DATA2= 32H, 4AH, 8BH (12)

CX= 0 CF= 1 (4)

Q3) [23 *points*] Write a program which:

(10) **a)** finds the **largest byte** from the set {18, 23, 32, 75, 64}

(3) **b)** stores the largest byte into another data area called "Largest"

(10) **c)** multiplies the largest byte by 4 which is entered from the keyboard.
{Hint: Use service 01H to get 4 from keyboard. Your program should check whether 4 is entered or not. If not, it should wait until 4 is entered}

```
        Format of service 01H:   MOV AH, 1
                                 INT 21H
```

```
DATA1    DB   18, 23, 32, 75, 64
LARGEST  DB   ?

         MOV CX, 5
         mov BX, Offset DATA1
         mov DL, 0
NEXT:    CMP DL, [BX]
         JA K1
         mov DL, [BX]
   K1:   INC BX
         LOOP NEXT
         mov DI, Offset LARGEST
         MOV [DI], DL
         ; Get character from keyboard

AGAIN:   MOV AH, 1
         INT 21H
         CMP AL, '4'
         JE K2
         JMP AGAIN
   K2:   MOV AL, 4        ; AX = AL x DL
                          ;            (4 x 75)
         MUL DL           ; ↙
```

Q4) [20 *points*] Find the contents of the following registers after the execution of following program segments. Show your work for each case.

a)

```
MOV AL, 83H
MOV CL, 2
ROR AL, CL
```

AL = 1000 0011   CF
1100 0001   1
1110 0000   1

AL= E0H   CF= 1

(Assume that initially CF=0)

b)

```
MOV AL, 83H
MOV CL, 2
RCR AL, CL
```

AL = 1000 0011   CF 0
0100 0001   1
1010 0000   1

AL= A0H   CF= 1

c)

```
MOV AL, 83H
MOV CL, 2
SHR AL, CL
```

AL = 1000 0011
0100 0001
0010 0000

AL= 20H

d)

```
MOV AL, 83H
MOV CL, 2
SAR AL, CL
```

AL = 1000 0011
1100 0001
1110 0000

AL= E0H

e)

```
    MOV AH, 14
    SUB AH, 10
    JC  K1
    NOT AH
K1: NEG AH
```

AH
14
14-10=4 = 0000 0100

$$\begin{array}{r} 1110 \\ -1010 \\ \hline \end{array} \rightarrow \begin{array}{r} {}^{1\,1}\\ 1110 \\ +0110 \\ \hline \textcircled{1}\,0100 \end{array}$$

CF

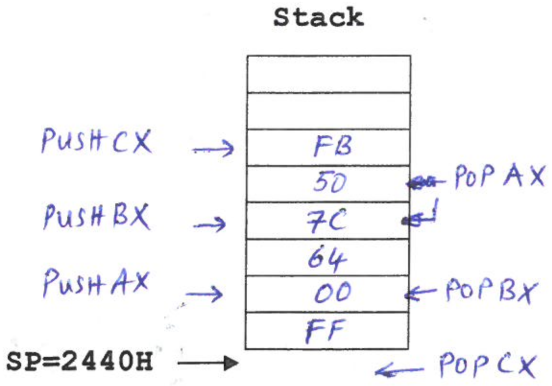
NEG AH = 1111 1100 → 0000 0000
+ 1111 1100

AH= FCH

f)

```
SUB AL, AL
MOV AH, 0FEH
ADD AH, 1
MOV BX, 647CH
MOV CX, 50FBH
PUSH AX
PUSH BX
PUSH CX
INC AX
DEC BX
POP AX
POP BX
POP CX
```

→ AH = FFH



AX= 50FBH   BX= 647CH   CX= FF00H

Q5) [20 *points*] Trace the following program and find the values stored into DATA2 and DATA3.

```
            .DATA
DATA1       DB    -16, 34, -57, -81
DATA2       DB    ?
DATA3       DB    ?
COUNT       EQU   4
            .CODE
            MOV AX, @DATA
            MOV DS, AX
            MOV CX, COUNT
            MOV BX, Offset DATA1
            MOV DX, 0
            CALL TASK
            MOV AH, 0ACH
            INT 21H
            END
            ;................................................
      TASK Proc
      K1:   MOV AL, [BX]
            CBW
            ADD DX, AX
            INC BX
            LOOP K1
            MOV AX, DX
            CWD
            MOV CX, COUNT
            IDIV CX
            NOT AX
            INC AX
            MOV DATA2, AX
            MOV DATA3, DX
            RET
            ENDP TASK
```

DATA2 : 30

DATA3 : 0