EE213 Computer Organization and Assembly Language
Quiz III– FALL 2018
December 07th, 2018
Section: B
Paper-B

Student Name: _____ Roll#_____

MOD=11			Effective Address Calculation			
R/M	W = 0	W = 1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	AH	SP	100	(SI)	(SI) + D8	(SI) + D16
101	CH	BP	101	(DI)	(DI) + D8	(DI) + D16
110	DH	SI	110	DIRECT ADDRESS	(BP) + D8	(BP) + D16
111	BH	DI	111	(BX)	(BX) + D8	(BX) + D16

ADD	0000 00DW
ADD reg/mem,imm	1000 000W (Ext 000)
MOV	1000 100DW
MOV reg/mem,imm	1100 011W (Ext 000)
MUL	1111 011W (Ext 100)
SUB	0010 10DW
SUB reg/mem,imm	1000 100W (Ext 101)
POP reg16	0101 1000
POP mem16	1000 1111 (Ext 000)
PUSH reg16	0101 0000
PUSH imm	0110 1000
PUSH mem16	1111 1111 (Ext 110)

1. Provide machine language (in hex-decimal) for the following x86 instructions **[14 Points]**

a. `MUL WORD PTR [BX+DI+08h]`
1111 0111 01 100 001 ←08
F7 61 08h

b. `MOV DX, 1F1Eh`
1100 0111 + 010 ←1E 1F
C7 + 2 ←1E 1F
C9 1E 1Fh

c. `ADD [BX+DI+1709h], 0F0E1h`
1000 0001 10 000 001 ← 09 17 ← E1 F0
81 81 09 17 E1 F0h

d. `SUB DX,CX`
0010 1001 11 001 010
29 CAh

e. `MUL CH`
1111 0110 + 101
F6 + 5
FBh

f. `SUB [DI], BX`
0010 1001 00 011 101
29 1Dh

g. `PUSH 170h`
0110 1000 ← 70 01
68 70 01h

2. Calculate the average of fourth column of the following 2D array in EDX **[4 Points]**

45	32	33	3	19	45
01	12	76	12	23	43
20	100	18	81	98	33

```
RowLength = 6
ColumnLength = 3
column_index = 3
.CODE
      MOV   ebx,OFFSET array
      ADD   ebx,(TYPE array*column_index)
      MOV   ecx,3
      MOV   eax,[ebx]
      L1:
      ADD   eax,[ebx+ Rowlength*TYPE array]
      ADD   ebx, Rowlength*TYPE array
      LOOP L1
      MOV   DX,0
      MOV   CX,Column_length
      DIV   CX
      MOVZX EDX, AX
```

Wtg: 02%

Write a procedure that should calculate and replace each of the following elements with their square roots without using LOOP, make use of **string primitive instructions**: **[2 Points]**

SQUARES WORD 4,9,16,25,36,49,64,81,100

Solution:

```
P1 PROC
      MOV   EDI, OFFSET squares
      MOV   ESI, EDI
      MOV   BX, 2
      MOV   ECX, LENGTHOF squares
      L1:
      MOV   DX,0
      LODSW
      DIV   BX
      CBW
      STOSW
      INC   BX
      DEC   ECX
      CMP   ECX,0
      JA    L1
      Ret
P1 ENDP
```