EE213 Computer Organization and Assembly Language
Quiz III – FALL 2018
December 7th, 2018
Section: D
Paper-A

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. `ADD CH,CL`
0000 0000 11 001 101
00 CDh

b. `MOV [BX+DI+1709h], 0F0E1h`
1100 0111 10 000 001←09 17←E1 F0
C7 81 09 17 E1 F0h

c. `MUL SI`
1111 0111 + 110
F7 + 6
0FDh

d. `SUB [BP+108h],CL`
0010 1000 10 001 110 ← 08 01
28 8E 08 01h

e. `POP BYTE PTR [BX+DI+1CEh]`
1000 1111 10 000 001 ←CE 01
8F 81 CE 01h

f. `SUB BX, 127h`
1000 1001 + 011 ← 27 01
89 + 3 ← 27 01
8C 27 01h

g. `PUSH 170Dh`
0110 1000 ← 0D 17
68 0D 17h

2. Elaborate the following directive **[2 Points]**

```
.model   MEDIUM, C, NEARSTACK
```

Answer: The directive creates multiple code segments and a single data segment where stack segment is maintained within the data segments. Using C calling the number of bytes, equal to size of passed arguments, are added to ESP in parent procedure to clean up the stack

3. Calculate the average of third and fourth row of 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
190	11	43	67	13	15

```
Rowlength = 6
row_index = 2
num_of_rows = 2

.CODE
   MOV      ebx,OFFSET array
   ADD      ebx,(Rowlength*TYPE array*row_index)
   MOV      ecx,RowLength*num_of_rows
   MOV      eax,[ebx]

   L1:
   ADD      eax,[ebx+TYPE array]
   ADD      ebx, TYPE array
   LOOP     L1

   MOV      DX,0
   MOV      CX,Rowlength*num_of_rows
   DIV      CX
   MOVZX    EDX,AX
```

4. Write a procedure that should calculate and replace each of the following NEGATIVE elements with their mathematical positive values without using LOOP, make use of string primitive instructions: **[4 Points]**

SQUARES SWORD 4,9,-16,25,36,-49,64, 81,-100,121

Solution:

```
P1 PROC
      MOV   EDI, OFFSET squares
      MOV   CX, LENGTHOF squares
      L1:   CMP CX,0
      JE    EX
      MOV   AX, [EDI]
      CMP   AX, 0
      JGE   continue
      NEG   AX
      continue:STOSW
      DEC   CX
      JMP   L1
      EX:   ret
P1 ENDP
```