EE213 Computer Organization and Assembly Language
Quiz III (GR I) – Spring 2019
Thursday, May 02, 2019
Paper-A-Solution
Student Name: _____ Roll#_____

1. Given the following recursive procedure, and that **EAX = 0F1h, EBP = C1EFh** and **ESP = F0F0h**, draw out the whole stack (and stack frames) with addresses, till after func1's first recursive call. No point will be awarded without correct addresses. **[06 points]**

```
main PROC LOCAL X:DWORD, Y:DWORD
0010   PUSH   EBP
0014   MOV    EBP, ESP
0018   MOV    X, 01h
001C   MOV    Y, 04h
0020   INVOKE func1,X,Y
0024   LEAVE
0028   RET
main   ENDP
```

```
func1 PROC, param1:DWord, param2:DWORD USES EAX
0900   ENTER  4, 1
0904   MOV    EAX, 0
0908   MOV    AX, param1
090C   ADD    AX, param1
0910   INC    param2
0914   INVOKE func1, param1, param2
0918   LEAVE
091C   RET
func1  ENDP
```

F0E4	01
F0E0	04
F0DC	0024
F0D8	F0F0
F0D4	
F0D0	0F1
F0CC	01
F0C8	05
F0C4	0918
F0C0	F0D8
F0BC	
F0B8	02

F0F0	C1EF	;EBP Pushed, EBP = F0F0h now
F0EC	01	;X (local of main)
F0E8	04	;Y(local of main)
F0E4	01	;param1 (DWORD)
F0E0	04	;param2(DWORD)
F0DC	0024	;return to main
F0D8	F0F0	;EBP PUSHED,EBP=F0D8 NOW

F0D4		;4-bytes reserved for local data
F0D0	0F1	;EAX pushed
F0CC	01	;param1(DWORD)
F0C8	05	;param2(DWORD)
F0C4	0918	;return to func1
F0C0	F0D8	;EBP PUSHED,EBP=F0C0 NOW
F0BC		;4-bytes reserved for local data
F0B8	02	;EAX Pushed

2. Write equivalent x86 assembly PROTOTYPE for the following C++ function: **[02 Points]**
int sample (short *, char,char*, char array[])

ANSER:

Sample PROTO, ptr1: PRT WORD, var1: BYTE, ptr2: PTR BYTE, ptr3: PTR BYTE

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

DEC	48h
ADD	0000 00DW (EXT 000)
ADD reg16/mem16, imm16	81h
CMP	0011 10DW (EXT 111)
SUB	1000 00DW (EXT 101)
SUB reg16/mem16, imm16	81h
MOV	1000 10DW (EXT 000)
PUSH reg16/reg32	50h
PUSH mem16/mem32	FFh (EXT 110)

3. Encode the following instructions, provide only the hex-decimal encoded values: **[4 Points]**

1. **CMP DX, AX**

   0011 1001 11 000 010
   = 39 C2h

2. **MOV DX, 1008h**
   1000 1011b + 010b ← 08 10h

8B + 2 ← 08 10h
= 8D 08 10

3. **DEC EBX**
   48h + 03h(EBX)
   = 4Bh

4. **ADD EBP, 1C1h**
   81h + 05h(EBP) ← C1 01 00 00h
   = 86 C1 01 00 00h