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

1. Given the following recursive procedure, and that **EAX = 10h, EBP = 9090h** and **ESP = 9CFFh**, 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:WORD, Y:BYTE
      PUSH  EBP
      MOV   ESP, EBP ;EBP = 9CFFh
      MOV   X, 01h
      MOV   Y, 04h
      INVOKE      func1,X,Y
      LEAVE
      RET
main  ENDP
```

```
func1 PROC, param1:Word, param2:byte
      ENTER 4, 1
      MOV   EAX, 0
      MOV   AX, param1
      ADD   AX, param1
      INC   param2
      INVOKE      func1, param1, param2
      LEAVE
      RET
func1 ENDP
```

Answer

9CF8h	01
9CF6h	04
9CF5h	ret(main)
9CF1h	9CFF
9CEDh	
9CE9h	01
9CE7h	05
9CE6h	ret(func1)
9CE2h	9CF1
9CDEh	

9CFF	9090	;EBP = 9CFF now
9CFB	01	;X (local of main)
9CF9	04	;Y (local of main)
9CF8	01	;param1 (word)
9CF6	04	;param2 (byte)
9CF5	ret(main)	;return to main
9CF1	9CFF	;EBP =9CF1 now
9CED	?	;4-bytes reserved for local data
9CE9	01	;param1(word)
9CE7	05	;param2(byte)
9CE6	ret(func1)	

9CE2	9CF1	;EBP=9CD3 now
9CDE	?	;4-bytes reserved for local data

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

Answer:

```
sample PROTO, var1: DWORD, ptr1: PTR DWORD, var2: BYTE, var3:WORD, ptr2:PTR WORD
```

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. **SUB DX, [1008h]**

   1000 0011 00 010 110
   = 83 16h

2. **CMP [BP + 1008h], DX**

   0011 1001 10 010 110
   39 96 ← 08 10
   = **39 96 08 10h**

3. **PUSH EBP**
   50 + 5
   = **55h**

4. **ADD EBX, 0FC1h**

   81 + 3 ← C1 0F 00 00
   = **84 C1 0F 00 00h**