May 20, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Major Exam II

First Semester (092)

Time: 1:00 PM-3:30 PM

Student Name : _____

Student ID. : _____

Question	Max Points	Score
Q1	28	
Q2	36	
Q3	12	
Q4	24	
Total	100	

Dr. Aiman El-Maleh

[28 Points]

(Q1) Fill the blank in each of the following:

(1) Assume that ESP=00000100H and the address of TEST is 0000300AH. After executing the instruction CALL TEST, the content of ESP=_____.

(2) Assume that ESP=00000100H. After executing the instruction RET 8, the content of ESP=_____.

(3) The code to Jump to label L1 if regiser AL bits 3 and 6 are 1 or bit 5 is zero is:

(4) Assuming that EAX=8765432CH and ECX=FEDBA7E4H, executing the instruction SHRD EAX, ECX, 16 will set EAX=_____ and ECX=_____.

(5) To multiply the signed content of register EAX by 33.25 without using multiplications instructions, we use the following instructions:

(6) Assuming that all variables are 32-bit signed integers, the assembly code implementing the following equation **var3 = (-5*var1)/(8*var2 -10)** is:

(7) Suppose that we have a 64-bit number stored in memory in the variable I defined as I Qword. The assembly code to multiply this number by 8 is:

(8) Given that MS-DOS packs the year, month, and day into 16 bits in register DX, where bits 0 to 4 store the day, bits 5 to 8 store the month and bits 9 to 15 store the year relative to 1980. Write assembly code to print the date in day, month and year. For example if DX=0010011001101010, it will print 10/3/1999:

[36 Points]

(Q2) Answer <u>SIX</u> out of the following questions. Show how you obtained your answer:

(i) Given the following definition in the data segment:

```
Array DWORD   0,  1,  2,  3,  4
      DWORD 10,11,12,13,14
      DWORD 20,21,22,23,24
      DWORD 30,31,32,33,34
      DWORD 40,41,42,43,44
```

Determine what will be displayed after executing the following code**:**

```
    mov ecx, lengthof Array
    xor esi, esi
Next:
    mov eax, lengthof Array
    mul esi
    shl eax, 2
    mov eax, Array[eax+esi*4]
    Call WriteDec
    Call CrLf
    inc esi
    loop Next
```

(ii) Determine what will be displayed after executing the following code**:**

```
    push 5
    push 4
    call MyProc

MyProc PROC
    push ebp
    mov ebp, esp
    sub esp, 4
    push eax
    mov DWORD PTR [ebp-4],10
    mov eax, [EBP + 8]
    sub [ebp-4], eax
    shl DWORD PTR [ebp-4], 2
    mov eax, [EBP + 12]
    add [ebp-4], eax
    shr DWORD PTR [ebp-4], 1
    mov eax, [ebp-4]
    call WriteDec
    pop eax
    mov esp, ebp
    pop ebp
    ret 8
MyProc ENDP
```

(iii) Given the following definition in the data segment:

Array DWord 17,-10,30,-40,4,-5,8

Determine what will be displayed after executing the following code**:**

```
    xor eax, eax
    mov esi, -1
    mov ecx, lengthof Array
L1:
    inc esi
    test  Array[esi*4], 8000h
    loopz L1
    jz done
    inc eax
    cmp ecx, 0
    jnz L1
done:
    call WriteDec
```

(iv) Determine what will be displayed after executing the following code**:**

```
     mov ecx, 5
     mov eax, 12
Next:
     cmp   eax, 7
     ja    default
     jmp   jumptable[eax*4]

case01:
    add eax, 9
    jmp done
case23:
    add eax, 7
    jmp done
case45:
    add eax, 3
    jmp done
case67:
    add eax, 4
    jmp done
default:
    inc eax
done:
    shr eax, 1
    loop Next
    call WriteDec
exit
jumptable DWORD    case01, case01, case23, case23, case45, case45, case67,
case67
```

(v) Determine what will be displayed after executing the following code**:**

```
        mov eax, 3
        call MyProc
        call WriteDec

MyProc Proc
        push ebx
        cmp eax, 0
        je done
        cmp eax, 1
        je done
        dec eax
        mov ebx, eax
        call MyProc
        xchg ebx, eax
        dec eax
        call MyProc
        add eax, ebx
done:
        pop ebx
        ret
MyProc Endp
```

(vi) Given the following declaration in the data segment:

MyNumber Byte 9 dup(0)

Determine what will be displayed after executing the following code**:**

```
        mov  ax, 0ABCDh
        xor  esi, esi
        mov  ecx, 8
L1:     rol  ax, 2
        mov  bx, ax
        and  bx, 3
        add  bl, '0'
        mov  MyNumber[esi], bl
        inc  esi
        loop L1

        lea edx, MyNumber
        call WriteString
```

(vii) Given the following declaration in the data segment:

MyNumber Byte '1','2','3',0

Determine what will be displayed after executing the following code**:**

```
        xor esi, esi
        mov   eax, 0
L1:     imul  eax, 16
        movzx edx, MyNumber[esi]
        sub   edx, '0'
        add   eax, edx
        inc   esi
        cmp   MyNumber[esi],0
        jne   L1

        dec esi
        mov  ecx, 0
        mov  ebx, 8
L2:     mov  edx, 0
        div  ebx
        add  dl, '0'
        mov MyNumber[esi], dl
        dec esi
        cmp  eax, 0
        jnz  L2

       lea edx, MyNumber
       call WriteString
```

[12 Points]

(Q3) Write a macro, **CMul**, to multiply the signed content of register EAX by a constant **n** passed as a aparmeter to the macro. The macro should be based on using shift and add instructions and should not use MUL or IMUL instructions. The macro should preserve the content of all temporary registers used.

[24 Points]

(Q4)

(i) Write a procedure, **BubbleSort**, to sort an array of integers in an **ascending** order. The number of integers to be sorted and the address of the array to be sorted are assumed to be passed on the stack. The procedure should maintain the content of all registers to their state before its execution.

The pseudocode for the **BublleSort** procedure is given below:

```
BubbleSort (ArrayPointer, ArraySize)
        pass = 1
        do {
                swap_occurs = 0
                for (i= 1 to ArraySize-pass)
                        if (Array[i-1] > Array[i])
                                swap ith and (i-1)th elements of the array
                                swap_occurs = 1
                        end if
                end for
                pass = pass+1
        while (swap_occurs && pass <= ArraySize -1)
end BubbleSort
```

(ii) Write a complete program, showing the place of procedure definition, to use the procedure **BubbleSort** to sort the Array given below:

Array DWord 10, 2, 0, 15, 25, 30, 7, 22

Note that the Content of Array after sorting will be:
Array DWord 0, 2, 7, 10, 15, 22, 25, 30