

**KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS**  
**COLLEGE OF COMPUTER SCIENCE AND ENGINEERING**  
***COMPUTER ENGINEERING DEPARTMENT***

**COE 205 - Term 061**  
**Assignment #3**

**Problem 1 (20 points)**

Consider the following:

```
.data
    Table1    DWORD 8, 10h, 9, 20h, 10, 23h, F6h, 22h, 16h
    NumE      EQU 9
    Result    DWORD ?

.code
    MOV     EBX, OFFSET Table1
    MOV     ECX, NumE
    MOV     ESI, 0
    MOV     EAX, 0
Lp1:    CMP     EAX, [EBX+ESI*4]        ; Compare values
        JAE     Cont                 ; Jump if Above or Equal
        MOV     EAX, [EBX+ESI*4]
Cont:   INC     ESI
        LOOP   Lp1
        MOV     Result, AX
```

1. What is the directive EQU used for?
2. What is the effect of the directive OFFSET?
3. What is the value of the variable Result after the execution of the program?
4. What does this program do?

## Problem 2 (20 points)

Consider the following:

```
.data
    Table1    BYTE 01, 03, 06, 12, 15, 16, 20, 23, 27, 31,
              34, 67
    NumE      EQU 12
              BYTE 10 DUP(?)
    Var1      BYTE 19
    Var2      BYTE 54

.code
    MOV     ESI, 0
Loop1:    MOV     AL, Table1[ESI]
          CMP     AL, Var1           ;Compare instruction
          JGE     Insert           ;Jump if Greater or Equal
          INC     ESI
          CMP     ESI, NumE         ;Compare Instruction
          JNZ     Loop1           ;Jump if Result Not Zero
Insert:   MOV     BL, Var1
          MOV     Table1[ESI], BL
Loop2:   INC     ESI
          MOV     BL, Table1[ESI]
          MOV     Table1[ESI], AL
          MOV     AL, BL
          CMP     ESI, NumE+1
          JNZ     Loop2
```

1. What does the first loop (Label **Loop1**) do? Explain.
2. What does the second loop (Label **Loop2**) do? Explain
3. What does the sequence between the label **Insert** and the label **Loop2** do? Explain.
4. What does this program do? Explain.

### Problem 3 (20 points)

Consider the following program.

```
.data
    Table BYTE 2, 3, 7, 1, 2, 5, 6
    NumE EQU   SIZEOF Table-1
    Var   BYTE ?

.code
    MOV   EBX, OFFSET Table
    MOV   ECX, NumE
    MOV   AL, 0
Lp1:    ADD   AL, [EBX]
        INC   EBX
        LOOP Lp1
        MOV   Var, AL
```

- What will be the value of AL after the execution of the program?
- What does this program do?
- Modify this program to take care of when the result is larger than 255 (byte maximum value). We suppose that the result would fit in a 32-bits word.

## Problem 4 (20 points)

Consider the following program:

```
.data
Table1    BYTE  5, 10, 15, 20, 22, 27, 30, 56, 2, 78
Table2    BYTE  67, 1, 78, 2, 89, 3, 95, 4, 76, 5
Table3    BYTE  SIZEOF Table1 DUP(0)

.code
        MOV  EBX, OFFSET Table1
        MOV  ESI, 0
        MOV  ECX, SIZEOF Table1
Lp1:    MOV  AL, [EBX+ESI]
        SUB  AL, [EBX+SIZEOF Table1+ESI]
        JGE  Write      ; Jump if Greater or Equal
        NEG  AL
Write : MOV  [EBX+SIZEOF Table1+SIZEOF Table2+ESI], AL
        INC  ESI
        LOOP Lp1
```

- a. What is the addressing mode [EBX+ESI] called?
  - A. Indexed
  - B. Based
  - C. Based-indexed
  
- a. What is the addressing mode [SIZEOF Table1+ESI] called?
  - A. Indexed
  - B. Based-indexed
  - C. Based-indexed with displacement
  
- b. What is the value of the address pointed to by [EBX+SIZEOIF Table1+ESI] when ESI =3
  
- c. What does this program do?

### **Problem 5 (20 points)**

Write a program that counts the number of even number in a defined table of bytes. The size of the array is arbitrary. An even number is a number that has the LSB bit equal to 0.

To find out whether a number is even or not, do the following:

- Move the number into AL
- Execute the instruction `AND AL, 01h`; which will make a bitwise logic AND between every bit in AL and the constant 01h which means will produce a result that has all bits at 0 except the LSB which can either be 0 or 1 depending whether AL contained an even or odd number.