# LAB 1

*Objective-3 : Find the Gray code of an 8-bit binary number.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
mov al, [1000h]
 mov bl,al
 shr al,01
 xor al,bl
mov [1001h],al
hlt
```

*Objective 4: Find the 2's complement of an 8 -bit number.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
mov al, [1000h]
 not al
 add al,01h
 mov [1001h],al
 hlt
```

# LAB 2

*Objective 2: Find the sum and average of N 16-bit numbers.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
        MOV SI,2000H
        MOV CL,[SI]
```

```
        MOV CH,00H
        MOV BX,CX
        MOV AX,0000H
L2:     INC SI
        INC SI
        ADD AX,[SI]
        JNC L1
        INC CH
L1:     DEC CL
        JNZ L2
        INC SI
        INC SI
        MOV [SI],AX
        INC SI
        INC SI
        MOV [SI],CH
        DIV BX
        INC SI
        INC SI
        MOV [SI],AX
        INC SI
        INC SI
        MOV [SI],DX
        HLT
```

## *Objective 3: Count no. of 0s in an 8-bit number.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
        MOV BX,2000H
        MOV AL,[BX]
```

```
        MOV CL,08H
        MOV CH,00H
L2:     SHR AL,1H
        JC L1
        INC CH
L1:     DEC CL
        JNZ L2
        INC BX
        MOV [BX],CH
        HLT
```

# LAB 3

*Objective-1 : Find the largest/smallest number (8-bit number) from a given array of size N.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
.data
 count db 04h
 value db 09h, 10h,05h,03h
res db ?
.code
 MAIN PROC
        mov ax, data
        mov ds, ax
       mov cl, count
       dec cl
       LEA SI, value
```

```
        mov al, [SI]
up:     inc si
        cmp al, [si]
        jnl nxt   # jnc for smallest
        mov al, [si]
nxt:    dec cl
        jnz up
        LEA DI, res
        mov [DI], al
END MAIN
```

# LAB 4

*Obj-1: Perform Addition and Subtraction of two 32-bit numbers using data processing addressing mode (with 8-bit immediate data).*

**Program**:

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
.global _start
 _start:
mov r0, #0x40
mov r1, #0x50
 adds r2,r0,#0x50
subs r3,r0,#0x50
mul r4,r0,r1
 my_exit: b my_exit
```

*Objective-3: Perform the logical operations (AND, OR, XOR, and NOT) on two 32-bit numbers using load/store addressing mode*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
 .global _start
 _start:
            LDR R0,=0X10100000
             LDR R1,[R0],#4
             LDR R2,[R0],#4
            ANDS R3,R2,R1
             STR R3,[R0],#4
             ORR R4,R2,R1
            STR R4,[R0],#4
            EOR R5,R2,R1
             STR R5,[R0],#4
             MVN R6, R1
            STR R6,[R0]
 my_exit: b my_exit
```

# LAB 5

*Objective-2 : Separate Even numbers and odds numbers in an array of size N using ARM Assembly language.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
.data
input:     .word 5, 12, 7, 8, 3, 9, 14
N:         .word 7
evenArr:   .space 28
oddArr:    .space 28
```

```
.text
.global _start
_start:
    LDR r0, =input
    LDR r1, =N
    LDR r1, [r1]
    LDR r2, =evenArr
    LDR r3, =oddArr
    MOV r4, #0
loop:
    CMP r4, r1
    BEQ done
    LDR r5, [r0, r4, LSL #2]
    AND r6, r5, #1
    CMP r6, #0
    BEQ store_even
store_odd:
    STR r5, [r3], #4
    B next
store_even:
    STR r5, [r2], #4
next:
    ADD r4, r4, #1
    B loop
done:
    B done
```

# LAB 6

*Objective - 2: Find the Fibonacci Series up to n digits.*

**Program:**

```
//NAME:- SK SHAKEEL AKHTAR
//REGD NO:- 2341001063
.global _start
_start:
    .text
    MOV R1, #1
    MOV R2, #0
    MOV R3, #0
    LDR R0, =COUNT
    LDR R6, =FIB_SERIES
    LDRB R5, [R0]
    CMP R5, #1
    BLE STOP1
    STRB R2, [R6], #1
    SUBS R5, R5, #1
    STRB R1, [R6], #1
BACK:
    SUBS R5, R5, #1
    BEQ STOP
    ADD R3, R1, R2
    STRB R3, [R6], #1
    MOV R2, R1
    MOV R1, R3
    B BACK
STOP1:
    STRB R3, [R6]
STOP:
    B .
    .data
COUNT:
    .byte 0x0A
```

```
FIB_SERIES:
    .byte 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0 @ 10-byte array
```