## COAL-LAB 06 (i22-1609)

### Task#1

model small

.stack 100h

.data

    arr db 5 dup(?)

    msg1 db 10,13,"Enter 5 Numbers in Array:$"

    msg2 db 10,13,"After Sorting Array:$"

.code

main proc

    mov ax, @data

    mov ds, ax

    ; Print "Enter 5 Numbers in Array:"

    mov dx, offset msg1

    mov ah, 09h

    int 21h

    mov cx, 5

    lea bx, arr

inputs:

    mov ah, 01h

    int 21h

    mov [bx], al

    inc bx

    loop inputs

    mov cx, 5

OuterLoop:

    mov bx, cx

    xor si, si

CompLoop:

    mov al, [arr+si]

    mov dl, [arr+si-1]

    cmp si, 0

    je noSwap   ; Jump if equal (si == 0)

    cmp al, dl

    jnc noSwap  ; Jump if not carry (al >= dl)

    xchg al, dl

    mov [arr+si], al

    mov [arr+si-1], dl

noSwap:

    inc si

    dec bx

    jnz CompLoop

    loop OuterLoop

    ; Print "After Sorting Array:"

    mov dx, offset msg2

    mov ah, 09h

    int 21h

    mov cx, 5

    lea bx, arr

Outputs:

    mov dl, [bx]

    mov ah, 02h

    int 21h

    mov dl, ' '

    int 21h

    inc bx

    loop Outputs

    mov ah, 4ch

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#2

.model small

.stack 100h

.data

array db 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100

.code

main proc

    mov ax, @data

    mov ds, ax

    mov cx, 100

    mov si, 0

L1:

    mov al, array[si]

    test al, 1     ; Check if the number is odd by testing the least significant bit

    jz skip        ; If even, skip printing

    add al, '0'    ; Convert the number to its ASCII representation

    mov dl, al

    mov ah, 2

    int 21h

skip:

    inc si

    loop L1

    mov ax, 4c00h

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#3

model small

.data

num db 2,1,3,4,5,6,5,7,9,9

msg db "Enter First Number: $"

msg2 db "Enter Second Number: $"

msg1 db 10,13, "Both numbers are Found $"

msg3 db 10,13, "One of the numbers is Found $"

msg4 db 10,13, "Neither number is Found $"

value1 db ?

value2 db ?

.code

main proc

    mov ax, @data

    mov ds, ax

    ; Input first number

    lea dx, msg

    mov ah, 09h

    int 21h

    mov ah, 01h

    int 21h

    sub al, '0'

    mov value1, al

    ; Input second number

    lea dx, msg2

    mov ah, 09h

    int 21h

    mov ah, 01h

    int 21h

    sub al, '0'

    mov value2, al

    ; Compare both numbers with array elements

    lea si, num

    mov cx, 10

CheckNumbers:

    mov bl, [si]

    cmp al, bl

    jz CheckSecondNumber

    inc si

    loop CheckNumbers

    ; If first number not found, check second number

    mov al, value2

    lea si, num

    mov cx, 10

CheckSecondNumber:

    mov bl, [si]

    cmp al, bl

    jz BothNumbersFound

    inc si

    loop CheckSecondNumber

    ; Neither number found

    mov dx, offset msg4

    jmp DisplayMessage

BothNumbersFound:

    mov dx, offset msg1

    jmp DisplayMessage

DisplayMessage:

    mov ah, 09h

    int 21h

    ; Exit the program

    mov ah, 4ch

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#4

.model small

.stack 100h

.data

var1 db ?

var2 db ?

    msg db 10,13, "Enter a Number: $"

    msg1 db 10,13, "Enter Second Number: $"

    msg2 db 10,13, "Second Number is Greater: $"

    msg3 db 10,13, "First Number is Greater: $"

.code

main proc

    mov ax, @data

    mov ds, ax

    mov dx, OFFSET msg

    mov ah, 09h

    int 21h

    mov ah, 01h

    int 21h

    sub al, 30h

    mov var1, al

    mov bl, al

    mov dx, OFFSET msg1

    mov ah, 09h

    int 21h

    mov ah, 01h

    int 21h

    sub al, 30h

    mov var2, al

    cmp al, bl

    jg greater

    jl lesser

    exit:

        mov ah, 4ch

        int 21h

    greater:

        mov dx, OFFSET msg2

        mov ah, 09h

        int 21h

        jmp exit

    lesser:

        mov dx, OFFSET msg3

        mov ah, 09h

        int 21h

        jmp exit

main endp

end main

A screenshot of a computer

Description automatically generated

### Task#5

.model small

.stack 100h

.data

    arr db 5 dup(?)

    msg1 db 10,13,"Enter 5 Numbers in Array:$"

    msg3 db 10,13,"Smallest Value: $"

    msg4 db 10,13,"Largest Value: $"

.code

main proc

    mov ax, @data

    mov ds, ax

    ; Print "Enter 5 Numbers in Array:"

    mov dx, offset msg1

    mov ah, 09h

    int 21h

    mov cx, 5

    lea bx, arr

inputs:

    mov ah, 01h

    int 21h

    mov [bx], al

    inc bx

    loop inputs

    mov cx, 5

OuterLoop:

    mov bx, cx

    xor si, si

CompLoop:

    mov al, [arr+si]

    mov dl, [arr+si-1]

    cmp si, 0

    je noSwap   ; Jump if equal (si == 0)

    cmp al, dl

    jnc noSwap  ; Jump if not carry (al >= dl)

    xchg al, dl

    mov [arr+si], al

    mov [arr+si-1], dl

noSwap:

    inc si

    dec bx

    jnz CompLoop

    loop OuterLoop

    ; Print the first value in the array

    mov dx, offset msg3

    mov ah, 09h

    int 21h

    mov dl, [arr] ; Get the first value

    mov ah, 02h

    int 21h

    ; Print the last value in the array

    mov dx, offset msg4

    mov ah, 09h

    int 21h

    mov dl, [arr+4] ; Get the last value

    mov ah, 02h

    int 21h

    mov ah, 4ch

    int 21h

main endp

end main

A screenshot of a computer

Description automatically generated