COAL LAB 6

ABDUL SAMI QASIM

22I-1725

CY-D

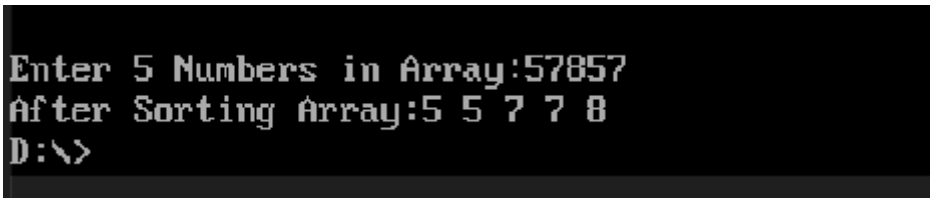
TASK1

CODE

```
.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
```

OUTPUT



TASK2

CODE

```
.model small
.stack 100h
```

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

   mov cx, 100 ; Number of elements in the array
   lea si, arr ; Load the address of the array into SI
   mov bx, 1   ; Starting number to check

   findOdds:
   mov ax, bx
   and ax, 1   ; Check the least significant bit of AX
   cmp ax, 1   ; Compare with 1 to check if odd
   jne notOdd  ; Jump if not equal (even number)

   mov [si], bl ; Store the odd number in the array
   inc si       ; Increment the array index

   notOdd:
   inc bx       ; Increment the number
   loop findOdds

   ; Print "Odd Numbers from 1 to 100:"
   mov dx, offset msg
   mov ah, 09h
   int 21h

   mov cx, 100 ; Number of elements in the array
   mov si, offset arr ; Load the offset of the array into SI

   printLoop:
   mov dl, [si] ; Load the current element from the array into DL
   add dl, 30h ; Convert the number to ASCII character
   mov ah, 02h ; Function to print character
   int 21h     ; Print the character

   inc si      ; Increment the array index
   loop printLoop

   mov ah, 4ch ; Exit program
   int 21h
main endp
end main
```

OUTPUT

```
Odd Numbers from 1 to 100: 13579;=?ACEGIKMOQSUWY[]_acegikmoqsuwy{}ªüâàçëïîìÅæôcde
fghijklmnopqrstuvwxyz{|}~ªÇüéâäàåçêëèïîìÄÅÉæÆôöò
D:\>
```

TASK3

CODE

```
.model small
.stack 100h
.data
    arr db 1, 1, 3, 1, 5, 6, 7, 1, 9, 0 ; Array of 10 elements
    num1 db ?
    num2 db ?
    result db 10 dup('$') ; Variable to store the result
    msg1 db 10,13,"num1: $"
    msg2 db 10,13,"num2: $"
    foundMsg db 10,13,"FOUND$"
    notFoundMsg db 10,13,"NOT FOUND$"
.code
main proc
    mov ax, @data
    mov ds, ax

    ; Take input for num1
    mov dx, offset msg1
    mov ah, 09h
    int 21h
    mov ah, 01h ; Function to read character
    int 21h     ; Read character from standard input
    sub al, '0' ; Convert ASCII to binary
    mov num1, al

    mov cx, 10 ; Number of elements in the array
    lea si, arr ; Load the address of the array into SI
    mov bl, num1 ; Load num2 into BL
    checkNum1:
    cmp bl, [si] ; Compare current element with num1
    je num1Found ; Jump if equal (num1 is found)
    inc si       ; Increment the array index
    loop checkNum1

    mov dx, offset notFoundMsg ; Load the offset of "NOT FOUND" message into DX
    mov ah, 09h ; Function to print string
    int 21h
    jmp input2


    num1Found:
    mov dx, offset foundMsg ; Load the offset of "FOUND" message into DX
    mov ah, 09h ; Function to print string
    int 21h



    ; Take input for num2
    input2:
    mov dx, offset msg2
    mov ah, 09h
    int 21h
    mov ah, 01h ; Function to read character
    int 21h     ; Read character from standard input
    sub al, 30h ; Convert ASCII to binary
    mov num2, al
```

```
    ; Check if num2 is in the array
    mov cx, 10 ; Number of elements in the array
    lea si, arr ; Load the address of the array into SI
    xor bl,bl
    mov bl, num2 ; Load num2 into BL

    checkNum2:
    cmp bl, [si] ; Compare current element with num2
    je num2Found ; Jump if equal (num2 is found)
    inc si       ; Increment the array index
    loop checkNum2

    ; If execution reaches here, both numbers are not found
    mov dx, offset notFoundMsg ; Load the offset of "NOT FOUND" message into DX
    mov ah, 09h ; Function to print string
    int 21h
    jmp endProgram



    num2Found:
    mov dx, offset foundMsg ; Load the offset of "FOUND" message into DX
    mov ah, 09h ; Function to print string
    int 21h


    endProgram:
    mov ah, 4ch ; Exit program
    int 21h
main endp
end main
```

OUTPUT

TASK4
CODE

```
.model small
.stack 100h
.data
    num1 db ?
    num2 db ?
    msg1 db 10,13,"num1: $"
    msg2 db 10,13,"num2: $"
```

```
    bigmsg db 10,13,"larger number: $"
.code
main proc
    mov ax, @data
    mov ds, ax

    ; Take input for num1
    mov dx, offset msg1
    mov ah, 09h
    int 21h
    mov ah, 01h ; Function to read character
    int 21h     ; Read character from standard input
    sub al, '0' ; Convert ASCII to binary
    mov num1, al

    mov dx, offset msg2
    mov ah, 09h
    int 21h
    mov ah, 01h ; Function to read character
    int 21h     ; Read character from standard input
    sub al, '0' ; Convert ASCII to binary
    mov num2, al

    mov al, num1
    cmp al, num2
    jbe secondnum

    mov dx, offset bigmsg
    mov ah, 09h
    int 21h
    mov al, num1
    add al, '0'
    mov dl, al
    mov ah, 02h
    int 21h

    jmp endProgram

secondnum:
    mov dx, offset bigmsg
    mov ah, 09h
    int 21h
    mov al, num2
    add al, '0'
    mov dl, al
    mov ah, 02h
    int 21h

endProgram:
    mov ah, 4ch ; Exit program
    int 21h
main endp
end main
```

OUTPUT

TASK5

CODE

```
.model small
.stack 100h
.data
    arr db 5 dup(?)
    msg1 db 10,13,"Enter 5 Numbers in Array:$"
    msg2 db 10,13,"After Sorting Array:$"
    msg3 db 10,13,"Smallest Number: $"
    msg4 db 10,13,"Largest Number: $"
.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


    mov dx, offset msg3
    mov ah, 09h
    int 21h

    mov dl, arr
    mov ah, 02h
    int 21h

    mov dx, offset msg4
    mov ah, 09h
    int 21h

    mov dl, arr+4
    mov ah, 02h
    int 21h

    mov ah, 4ch
    int 21h
main endp
end main
```

OUTPUT