**Nawaiz copyright claim**

**Program 1**

org 100h

section .data

buffer\_size equ 255 ; Define buffer size for input

buffer resb buffer\_size ; Reserve space for input buffer

newline db 0Ah, 0Dh ; Newline characters for formatting

section .code

mov ah, 09h ; DOS function: Display string

lea dx, prompt ; Load address of prompt string

int 21h ; Call DOS

mov ah, 0Ah ; DOS function: Buffered input

lea dx, buffer ; Load address of input buffer

int 21h ; Call DOS

mov ah, 09h ; DOS function: Display string

lea dx, newline ; Load address of newline string for formatting

int 21h ; Call DOS

mov ah, 09h ; DOS function: Display string

lea dx, buffer ; Load address of input buffer

int 21h ; Call DOS

mov ah, 4Ch ; DOS function: Exit program

int 21h ; Call DOS

section .data

prompt db 'Enter a string: $' ; Prompt string

**Program 2**

.model small

.stack 100h

include 'emu8086.inc'

.data

buffer\_size equ 255 ; Define buffer size for input

buffer resb buffer\_size ; Reserve space for input buffer

newline db 0Ah, 0Dh ; Newline characters for formatting

.code

mov ax, @data

mov ds, ax

main proc

; Display prompt

mov ah, 09h ; DOS function: Display string

lea dx, prompt ; Load address of prompt string

int 21h ; Call DOS

; Read input from the user

mov ah, 0Ah ; DOS function: Buffered input

lea dx, buffer ; Load address of input buffer

int 21h ; Call DOS

; Display reversed string

lea si, buffer ; Load source index with the address of the input buffer

mov di, si ; Set destination index to the same address initially

add di, buffer\_size ; Move destination index to the end of the buffer

reverse\_loop:

dec di ; Move the destination index backward

mov al, [si] ; Load a character from the source

mov [di], al ; Store the character in reverse order

inc si ; Move the source index forward

cmp al, '$' ; Check for the end of the string

jne reverse\_loop ; Continue the loop if not the end

; Display the reversed string

mov ah, 09h ; DOS function: Display string

lea dx, buffer ; Load address of reversed string

int 21h ; Call DOS

; Exit the program

mov ah, 4Ch ; DOS function: Exit program

int 21h ; Call DOS

main endp

section .data

prompt db 'Enter a string: $' ; Prompt string

**Program 3**

org 100h

include 'emu8086.inc'

.data

num1 db ? ; Declare variable for the first number

num2 db ? ; Declare variable for the second number

num3 db ? ; Declare variable for the third number

temp db ? ; Declare a temporary variable for swapping

.code

main proc

mov ax, @data ; Initialize the data segment

mov ds, ax ; Set ds to the data segment

print "Enter the first number: "

mov ah, 01 ; DOS function: Read character from standard input

int 21h ; Call DOS

sub al, 30h ; Convert ASCII character to integer ('0' -> 0)

mov num1, al ; Store the first number

call N\_line ; Move to the next line

print "Enter the second number: "

mov ah, 01 ; DOS function: Read character from standard input

int 21h ; Call DOS

sub al, 30h ; Convert ASCII character to integer ('0' -> 0)

mov num2, al ; Store the second number

call N\_line ; Move to the next line

print "Enter the third number: "

mov ah, 01 ; DOS function: Read character from standard input

int 21h ; Call DOS

sub al, 30h ; Convert ASCII character to integer ('0' -> 0)

mov num3, al ; Store the third number

; Swap num1 and num2

mov al, num1 ; Move num1 to AL

mov temp, al ; Store num1 in the temporary variable

mov al, num2 ; Move num2 to AL

mov num1, al ; Store num2 in num1

mov al, temp ; Move the original num1 from the temporary variable to AL

mov num2, al ; Store the original num1 in num2

; Swap num2 and num3

mov al, num2 ; Move num2 to AL

mov temp, al ; Store num2 in the temporary variable

mov al, num3 ; Move num3 to AL

mov num2, al ; Store num3 in num2

mov al, temp ; Move the original num2 from the temporary variable to AL

mov num3, al ; Store the original num2 in num3

; Swap num1 and num3

mov al, num1 ; Move num1 to AL

mov temp, al ; Store num1 in the temporary variable

mov al, num3 ; Move num3 to AL

mov num1, al ; Store num3 in num1

mov al, temp ; Move the original num1 from the temporary variable to AL

mov num3, al ; Store the original num1 in num3

print "After swapping, the numbers are: "

; Print num1

mov dl, num1 ; Move num1 to DL

add dl, 30h ; Convert num1 to ASCII

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

; Print num2

mov dl, num2 ; Move num2 to DL

add dl, 30h ; Convert num2 to ASCII

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

; Print num3

mov dl, num3 ; Move num3 to DL

add dl, 30h ; Convert num3 to ASCII

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

int 20h ; DOS function: Terminate program

main endp

N\_line proc

mov dl, 10 ; ASCII code for newline

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

mov dl, 13 ; ASCII code for carriage return

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

ret

N\_line endp

**Program 4**

org 100h

include 'emu8086.inc'

.data

array\_size dw 10 ; Length of the array

array dw 10 dup(?) ; Array to store integers

user\_number dw ? ; User input to check in the array

.code

main proc

mov ax, @data ; Initialize the data segment

mov ds, ax ; Set ds to the data segment

; Take array input from user

mov cx, array\_size ; Set the counter to the array size

mov si, offset array ; Set source index to the array

call InputArray ; Call subroutine to input array elements

; Take user input to check in the array

print "Enter a number to check: "

mov ah, 01 ; DOS function: Read character from standard input

int 21h ; Call DOS

sub al, 30h ; Convert ASCII character to integer

mov user\_number, ax ; Store user input in user\_number

; Check if user\_number is present in the array

mov cx, array\_size ; Set the counter to the array size

mov si, offset array ; Set source index to the array

call SearchArray ; Call subroutine to search for the number

; Display result

cmp ax, -1 ; Check if the number is not found

je not\_found

print "Number found in the array!"

jmp exit

not\_found:

print "Number not found in the array!"

exit:

int 20h ; DOS function: Terminate program

main endp

; Subroutine to input array elements

InputArray proc

A:

print "Enter an integer: "

mov ah, 01 ; DOS function: Read character from standard input

int 21h ; Call DOS

sub al, 30h ; Convert ASCII character to integer

mov [si], al ; Store the integer in the array

inc si ; Move to the next element

call N\_line ; Move to the next line

loop A

ret

InputArray endp

; Subroutine to search for a number in the array

SearchArray proc

mov bx, user\_number ; Number to search

mov di, -1 ; Initialize index to -1 (not found)

B:

inc di ; Move to the next element

cmp bx, [si] ; Compare user\_number with the current array element

je found ; If equal, jump to the found label

loop B

jmp not\_found ; If the loop completes, the number is not found

found:

mov ax, di ; Return the index where the number is found

ret

not\_found:

mov ax, -1 ; Return -1 if the number is not found

ret

SearchArray endp

N\_line proc

mov dl, 10 ; ASCII code for newline

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

mov dl, 13 ; ASCII code for carriage return

mov ah, 02 ; DOS function: Write character to standard output

int 21h ; Call DOS

ret

N\_line endp