Q#1 Program to ADD two numbers

```
section .data
    num1 db 5
    num2 db 3
    result db 0
    result_msg db 'Result: $'

section .bss
    result_str resb 4

section .text
    global _start

_start:
    mov al, [num1]
    add al, [num2]
    mov [result], al

    mov ax, [result]
    add ax, 48
    mov [result_str], al

    mov ah, 0x09
    lea dx, [result_msg]
    int 0x21

    mov ah, 0x09
    lea dx, [result_str]
    int 0x21

    mov ah, 0x4C
    int 0x21
```

Q#2 Program to SUB two numbers

```
section .data
    num1 db 10
    num2 db 4
    result db 0
    result_msg db 'Result: $'

section .bss
    result_str resb 4

section .text
    global _start

_start:
    mov al, [num1]
    sub al, [num2]
    mov [result], al

    mov ax, [result]
    add ax, 48
    mov [result_str], al

    mov ah, 0x09
    lea dx, [result_msg]
    int 0x21

    mov ah, 0x09
    lea dx, [result_str]
    int 0x21

    mov ah, 0x4C
    int 0x21
```

Q#3 Input two numbers and ADD them

```
section .data
    prompt_msg db 'Enter a number (0-9): $'
    result_msg db 0x0D, 0x0A, 'Result: $'
    num1 db 0
    num2 db 0
    result db 0

section .bss
    result_str resb 4

section .text
    global _start

_start:
    ; Prompt for first number
    mov ah, 0x09
    lea dx, [prompt_msg]
    int 0x21
```

```
; Read first number
mov ah, 0x01
int 0x21
sub al, 48
mov [num1], al

; Prompt for second number
mov ah, 0x09
lea dx, [prompt_msg]
int 0x21

; Read second number
mov ah, 0x01
int 0x21
sub al, 48
mov [num2], al

; Add numbers
mov al, [num1]
add al, [num2]
mov [result], al

; Convert result to ASCII
mov al, [result]
add al, 48
mov [result_str], al

; Print the result message
mov ah, 0x09
```

```
    lea dx, [result_msg]
    int 0x21

    ; Print the result
    mov ah, 0x09
    lea dx, [result_str]
    int 0x21

    ; Exit the program
    mov ah, 0x4C
    int 0x21
```

Q#4 Using a variable and enter key, write a code of your own idea

```
section .data
    prompt_name db 'Enter your name: $'
    prompt_age db 'Enter your age: $'
    output_msg db 0x0D, 0x0A, 'Hello, ', 0, '! You are ', 0, ' years old.$'
    name db 20 dup(0)  ; Reserve space for the name (up to 20 characters)
    age db 0           ; Variable to store age

section .text
    global _start

_start:
    ; Prompt for name
    mov ah, 0x09
    lea dx, [prompt_name]
    int 0x21
    ; Read name
    mov ah, 0x0A
    lea dx, [name]
    int 0x21

    ; Prompt for age
    mov ah, 0x09
    lea dx, [prompt_age]
    int 0x21

    ; Read age
    mov ah, 0x01
    int 0x21
    sub al, 48            ; Convert from ASCII to integer
    mov [age], al

    ; Prepare output message
    mov ah, 0x09
    lea dx, [output_msg]
    int 0x21

    ; Print name
    mov si, name + 1      ; Skip the first byte (length of the name)
    mov bx, 0             ; Reset counter for name output

print_name:
    mov al, [si]
    cmp al, 0
    je print_age
    mov ah, 0x0E        ; BIOS function to print character
    int 0x10
    inc si
    jmp print_name

print_age:
    ; Print ' years old.'
    mov ah, 0x0E
    mov al, ' '
    int 0x10
    mov al, [age]
    add al, 48
    int 0x10

    ; Exit the program
    mov ah, 0x4C
    int 0x21
```