QUESTION 1

```
.model small
.stack 100h
.data
    alph db "A B C D E F G H I J K L M N O P Q R S T U V X Y Z $"
.code
    main proc
      mov ax, @data
      mov ds, ax
      mov dx,offset alph
      mov ah, 09h
      int 21h

      mov ah, 2
      int 21h

      mov ah,4ch
      int 21h
      main endp

      end main
```

QUESTION 2

```
.model small
.stack 100h
.data
   numbers dw 0, 1, 2, 3, 4, 5, 6, 7, 8, 9  ; Array of numbers from 0 to 9
   count   dw 10                        ; Number of elements in the array
.code
main proc
```

```
    ; Initialize the data segment
    mov ax, @data
    mov ds, ax

    ; Initialize registers
    xor bx, bx          ; Clear BX to use it for the sum
    mov cx, count       ; Load the count of numbers into CX
    lea si, numbers     ; Load the address of the numbers array into SI

sum_loop:
    add bx, [si]        ; Add the current number to BX
    add si, 2           ; Move to the next number (each number is 2 bytes)
    loop sum_loop       ; Repeat until CX is 0

    ; Calculate the average
    mov ax, bx          ; Move the sum into AX
    xor dx, dx          ; Clear DX before division
    mov cx, count       ; Load count into CX
    div cx              ; AX = AX / CX, quotient in AX (average), remainder in DX

    ; Store the result (average) in BX
    mov bx, ax          ; Store the average in BX

    ; Optional: Display the average (not part of the requirements)
    ; Convert the average to ASCII and print if needed...

    ; Exit program
    mov ax, 4C00h       ; DOS terminate program
    int 21h
```

```
main endp
end main
```

QUESTION 3

```
.model small
.stack 100h
.data
    alph db "z y x w v u t s r q p o n m l k j i h g f e d c b a $"
.code
    main proc
      mov ax, @data
      mov ds, ax
      mov dx,offset alph
      mov ah, 09h
      int 21h

      mov ah, 2
      int 21h

      mov ah,4ch
      int 21h
      main endp

      end main
```

Question 4

```
.model small
.stack 100h
.data
   numbers db 5 dup(0)
   prompt db 'Enter a number (0-9): $'
```

```
    newline db 0Dh, 0Ah, '$'
    display_msg db 'You entered: $'

.code
main proc
    mov ax, @data
    mov ds, ax

    ; Input loop
    mov cx, 5
    mov si, 0

input_loop:
    ; Display prompt
    mov dx, offset prompt
    mov ah, 09h
    int 21h

    ; Read input
    mov ah, 01h
    int 21h
    sub al, '0'
    mov numbers[si], al

    ; Print newline
    mov dx, offset newline
    mov ah, 09h
    int 21h
```

```
    inc si
    loop input_loop

    ; Display numbers
    mov dx, offset display_msg
    mov ah, 09h
    int 21h

    mov cx, 5
    mov si, 0

display_loop:
    mov al, numbers[si]
    add al, '0'
    mov ah, 0Eh
    int 10h

    mov al, ' '
    mov ah, 0Eh
    int 10h

    inc si
    loop display_loop

    ; Print newline
    mov dx, offset newline
    mov ah, 09h
    int 21h
```

```
    ; Exit program
    mov ah, 4Ch
    int 21h

main endp
end main
```

Create Adding 3 number using Macro

```
;ex1
;using macro
;macro

.model  small
.stack 100h
.data
var1 dw 1    ; Declearing numbers
var2 dw 1
var3 dw 2
nextlinne db 0Dh, 0Ah,'$'
.code
display macro var1,var2,var3
    mov ax,var1  ;moving numbers to register ax
    add ax,var2  ;add 4 number in the reister
    add ax,var3  ;add 3 number in the reister

    mov dx, ax   ; for the display
    add dx,48    ; ASCII
    mov ah,2     ; For display the result
    int 21h
endm
```

```
    main proc
      mov ax,@data
      mov ds, ax              ;using ax
      display var1, var2,var3    ;calling macro

       main endp
end main
```