To take input a number:

```
MOV AH, 1H    ; calling input function
INT 21H
MOV BH, AL    ; fetch input in BH register
```

To show a number in output:

```
MOV AH, 2H    ; calling output function
MOV DL, BH    ; fetch data from BH to DL
INT 21H
```

To declare and to initialize variables:

```
.DATA
A DB 5    ; A is a variable, initial value is 5
B DB ?    ; B is a variable, not initialized

MAIN PROC
    MOV AX, @DATA    ; intializing DATA segment
    MOV DS, AX
```

To output a string:

```
.DATA
TEXT DB 'HELLO WORLD!$'
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    LEA DX, TEXT
    MOV AH, 9H    ; output string function
    INT 21H
```

Nayeem Mahmood / 4AM / C161026

New Line Function :

```
MOV AH, 2H
MOV DL, 0AH
INT 21H
```

Carriage Return Function :

```
MOV AH, 2H
MOV DL, 0DH
INT 21H
```

Beep Function :

```
MOV AH, 2H
MOV DL, 07H
INT 21H
```

Nayeem Mahmood / 4AM / C161026

Write an assembly program to print the two strings as showed below:

I am proud to read at IIUC.
I love studying assembly language.

```
.model small
.stack 100h
.data
text1 db "I am proud to read at IIUC.$"
text2 db "I love studying assembly language.$"
.code
main proc
    mov ax, @data
    mov ds, ax

    lea dx, text1
    mov ah, 9h
    int 21h

    mov ah, 2h
    mov dl, 0ah
    int 21h
    mov dl, 0dh
    int 21h

    lea dx, text2
    mov ah, 9h
    int 21h

    mov ah, 4ch
    int 21h

    main endp

end main
```

Nayeem Mahmood / 4AM / C161026

Write down an assembly program to convert from lower case to upper case letter and vice versa:

```
.MODEL SMALL
.STACK 100H
.DATA
TEXT1 DB "Enter a lower case letter: $"
TEXT2 DB "Enter an upper case letter: $"
.CODE
MAIN PROC
        MOV AX, @DATA
        MOV DS, AX

        LEA DX, TEXT1
        MOV AH, 9H
        INT 21H

        MOV AH, 1H
        INT 21H
        MOV BH, AL

        SUB BH, 20H  ; SUB BH, 32
                     ; This will work too


        MOV AH, 2H
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H

        MOV DL, BH
        INT 21H

        MOV AH,2H
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
```

Nayeem Mahmood / 4AM / C161026

```
LEA DX, TEXT2
MOV AH,9H
INT 21H

MOV AH, 1H
INT 21H
MOV BH, AL

ADD BH, 20H

MOV AH, 2H
MOV DL, 0AH
INT 21H
MOV DL, 0DH
INT 21H

MOV DL, BH
INT 21H

MOV AH, 4CH
INT 21H

MAIN ENDP

END MAIN
```

Nayeem Mahmood / 4AM / C161026

Write down an assembly program to print all the capital letters using FOR loop:

```
.MODEL SMALL
.STACK 100H
.CODE
MAIN PROC
        MOV CX, 26
        MOV AH, 2H
        MOV DL, 'A'

        PRINT:
        INT 21H
        INC DL
        MOV BL, DL
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
        MOV DL, BL
        LOOP PRINT

        MOV AH, 4CH
        INT 21H
        MAIN ENDP
END MAIN
```

Nayeem Mahmood / 4AM / C161026

Write down an assembly Program to Print a String "I love to eat Biriyani! $" 20 times using a for loop.

```
.MODEL SMALL
.STACK 100H
.DATA
TEXT DB "I love to eat Biriyani! $"
.CODE
MAIN PROC
        MOV AX, @DATA
        MOV DS, AX

        MOV CX, 20
        LEA DX, TEXT
        MOV AH, 9H

        PRINT:
        INT 21H
        MOV AH, 2H
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
        LEA DX, TEXT
        MOV AH, 9H
        LOOP PRINT

        MOV AH, 4CH
        INT 21H

        MAIN ENDP

END MAIN
```

Nayeem Mahmood / 4AM / C161026

Write down an assembly program using a user-defined level to print 80 stars (*).

```
.model small
.stack 100h
.code
main proc
        jmp userlevel

        userlevel:
        mov cx, 80
        mov ah, 2h
        mov dl, '*'
        print:
        int 21h
        loop print

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

Nayeem Mahmood / 4AM / C161026

Write down an assembly program that takes two inputs and add those inputs to produce the output.

```
.MODEL SMALL
.STACK 100H
.DATA
TEXT1 DB "Enter 1st number: $"
TEXT2 DB 0AH, 0DH, "Enter 2nd number: $"
TEXT3 DB 0AH, 0DH, "Sum = $"
A DB ?
B DB ?

.CODE
MAIN PROC
        MOV AX, @DATA
        MOV DS, AX

        LEA DX, TEXT1
        MOV AH, 9H
        INT 21H

        MOV AH, 1H
        INT 21H
        SUB AL, 48
        MOV A, AL

        LEA DX TEXT2
        MOV AH, 9H
        INT 21H

        MOV AH, 1H
        INT 21H
        SUB AL, 48
        MOV B, AL

        ADD AL, A
        MOV AH, 0
        AAA             ; adjust after addition
```

Nayeem Mahmood / 4AM / C161026

```
ADD AH, 48
ADD AL, 48

MOV BX, AX

MOV AH,9H
LEA DX, TEXT3
INT 21H

MOV AH, 2H
MOV DL, BH
INT 21H
MOV DL, BL
INT 21H

MAIN ENDP

END MAIN
```

Nayeem Mahmood / 4AM / C161026

Write an assembly program to find out biggest between two numbers.

```
.model small
.stack 100h
.data
text1 db "Enter first number: $"
text2 db 0AH, 0DH, "Enter second number: $"
text3 db 0AH, 0DH, "Largest = $"
.code
main proc
        mov ax, @data
        mov ds, ax

        lea dx, text1
        mov ah, 9h
        int 21h

        mov ah, 1h
        int 21h
        mov bh, al

        lea dx, text2
        mov ah, 9h
        int 21h

        mov ah, 1h
        int 21h
        mov bl, al

        lea dx, text3
        mov ah, 9h
        int 21h

        cmp bh, bl
        jg level1 ; if bh is greater, level1 will be executed
        jmp level2 ; else level2 will be executed
```

Nayeem Mahmood / 4AM / C161026

```
level1:
mov ah, 2h
mov dl, bh
int 21h
jmp exit

level2:
mov ah, 2h
mov dl, bl
int 21h
jmp exit

exit:
mov ah, 4ch
int 21h

main endp
end main
```

Nayeem Mahmood / 4AM / C161026