Task-1

```
.MODEL SMALL
.STACK 100H
.DATA
    l db ?
    b db ?
    r db ?
    msg1 db "Enter Length:$"
    str db 8 DUP(?)

    spring_msg db "Spring 20$"
    fall_msg db "Fall 20$"
    summer_msg db "Summer 20$"

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

    ; enter your code here
    mov ah, 9
    mov dx, offset msg1
    int 21h

    mov ah, 1
    int 21h
    sub al, 48
    add al, 1
    mov l, al

    mov ah, 2
    mov dl, 10
    int 21h
    mov ah, 2
    mov dl, 13
    int 21h

    ; loop
    mov cl, l
    mov si, 0
    loopstart:
      mov ah, 1
```

```
        int 21h
        cmp al, 13 ; check for Enter key
        je end_input
        mov str[si], al
        inc si
        dec cl
        jnz loopstart

    end_input:

    ; ================================================

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

    ; NEW LINE PRINT
    MOV DL, 0DH
    MOV AH, 2
    INT 21H

    ; ================================================

output:
    mov cl, l
    mov ah, 2
    mov si, 0
    start:
        mov dl, str[si]
        int 21h
        inc si
        dec cl
        jnz start

    ; Determine semester
    mov al, str[2]
    sub al, 30h  ; convert ASCII to integer
    mov r, al
    cmp r, 1
    je spring
    cmp r, 2
    je fall
    cmp r, 3
    je summer
```

```
spring:
    lea dx, spring_msg
    jmp print_output

fall:
    lea dx, fall_msg
    jmp print_output

summer:
    lea dx, summer_msg

print_output:
    ; Print semester
    mov ah, 9
    int 21h

    ; Print year
    mov ah, 2
    mov dl, '2'
    int 21h

    mov cl, 2  ; assuming the year is 2 digits
    mov si, 6   ; starting from the 6th index in the array
    print_year:
        mov dl, str[si]
        int 21h
        inc si
        dec cl
        jnz print_year

    jmp exit

exit:
    ; exit to DOS
    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN
```

Task-2

```
.MODEL SMALL

.STACK 100H
.DATA

msg1 db "Enter Length:$"

input db ?

.CODE
MAIN PROC

    MOV AX, @DATA
    MOV DS, AX

    mov ah, 9
    mov dl, offset msg1
    int 21h

    MOV AH,1
    INT 21H
    sub al,30h

    MOV input , AL


    ;================================

    MOV DL , 0AH
    MOV AH,2
    INT 21H
                ; NEW LINE PRINT
    MOV DL , 0DH
    MOV AH,2
    INT 21H
    ;============================

    mov ah,0
    mov al,input

    mov cx,ax
```

```
    insert:
    mov ah,1
    int 21h
    push ax
    loop insert

    ;================================

    MOV DL , 0AH
    MOV AH,2
    INT 21H
                    ; NEW LINE PRINT
    MOV DL , 0DH
    MOV AH,2
    INT 21H
    ;==============================

    mov ah,0
    mov al,input

    mov cx,ax

    mov cx,ax
    reverse:
    pop dx

    sub dx,20h

    mov ah,2
    int 21h
    loop reverse


MAIN ENDP
END MAIN
```