1.Program to print A to Z in small and capital letters using loop

```
.model small
.stack 100h
.data
.code
main proc
mov cx, 26    counter
mov ah, 2
mov dl, 65

L1:
int 21h
inc dl

loop L1
mov dl, 10
mov ah,2
INT 21h

mov dl, 13
mov ah,2
INT 21h

mov cx, 26
mov dl, 97
mov ah, 2
L2:
int 21h

inc dl
loop L2

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

2. Program to input a capital letter from user and convert it into small letter (uppercase to lowercase)

```
.model small
.stack 100h

.data
```

```
.code
main proc
mov ah, 1      ; input
int 21h
mov dl, al
add dl,32
mov ah, 2
int 21h

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

A ω5 ω 97

3. Program to input a small letter from user and convert it into capital letter (lowercase to uppercase)

```
.model small
.stack 100h
.data
.code
main proc
mov ah, 1
int 21h
mov dl, al
sub dl,32
mov ah, 2
int 21h

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

4. Program to input String from user and print its length

```
.model small
.stack 100h
.data
var1 db 100 dup("$")
.code
main proc
mov ax,@data
mov ds,ax
mov bl, 0                ; counts the length of string
```

```asm
        mov si,offset var1
l1:
        mov ah,1
        int 21h

        cmp al,13
        je printString
        mov [si],al
        inc si
        inc bl
        jmp l1

printString:
        mov dl, bl
        mov ah,2
        add dl, 48
        int 21h

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

5. Program to reverse the hard coded string

```asm
;dosseg
.model small
.stack 100h
.data
arr1 db 'k','c','n','g'
.code
main proc
mov ax,@data
mov ds,ax

mov si, offset arr1
mov cx, 4

L1:
mov ax, [si]
push ax
inc si
loop L1

mov cx, 4

L2:
```

```asm
    pop dx
    mov ah, 2
    int 21h
    inc si
    loop L2

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

6. Program to Input string from user and reverse it

```asm
;program to input string from user reverse it
.model small
.stack 100h
.data
var1 db 100 dup("$")
.code
inputString proc
mov ax,@data
mov ds,ax
mov bl, 0                      ; counts the length of string
mov si,offset var1
l1:
 mov ah,1
int 21h
cmp al,13
je printString
mov [si],al
inc si
inc bl
jmp l1

printString:
mov cl, bl
print:
dec si
mov dx,[si]
mov ah,2
int 21h
loop print

mov ah,4ch
int 21h
```

inputString endp
end inputString

7. Program to print the following pattern;

```
1
22
333
4444
55555
```
When user input 5

```
.model small
.stack 100h
.data
.code
main proc
mov ax,@data
mov ds,ax

mov ah, 1
int 21h
mov cl, al
sub cl, 48

mov dl,10
mov ah, 2
int 21h
mov dl,13
mov ah, 2
int 21h
mov bx, 1

L1:
push cx
mov cx, bx

L2:
Mov dx, bx
add dx, 48
mov ah,2
int 21h
loop L2

mov dl,10
```

```
mov ah, 2
int 21h
mov dl,13
mov ah, 2
int 21h
inc bl

pop cx
loop L1
mov ah,4ch
int 21h
main endp
end main
```

8.Program to convert lower case string to Upper case string

```
.model small
.stack 100h
.data
var1 db 100 dup("$")
.code
inputString proc
mov ax,@data
mov ds,ax
mov bl, 0                          ; counts the length of string
mov si,offset var1

l1:
mov ah,1
int 21h
cmp al,13
je printString
mov [si],al
inc si
inc bl
jmp l1
printString:

mov si,offset var1
mov cl, bl
uppercase:
mov dx,[si]
sub dx, 32
mov ah,2
int 21h
```

```
        inc si
        loop uppercase


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

9. Program to convert upper case string to lower case string

```
.model small
.stack 100h
.data
var1 db 100 dup("$")
.code
inputString proc
mov ax,@data
mov ds,ax
mov bl, 0                        ; counts the length of string
mov si,offset var1
l1:
mov ah,1
int 21h
cmp al,13
je printString

mov [si],al
inc si
inc bl
jmp l1

printString:
mov si,offset var1
mov cl, bl
lowercase:
mov dx,[si]
add dx, 32
mov ah,2
int 21h
inc si
loop lowercase

mov ah,4ch
int 21h
inputString endp
```

end inputString

10.Program to check the input number is Negative or Positive

```
.model small
.stack 100h
.data
num db 10 dup('$')
msgNeg db 'Given Number is Negative. $'
msgPos db 'Given Number is Positive. $'
.code
main proc
mov ax,@data
mov ds,ax

mov si, offset num
inputString:
mov ah, 1
int 21h
cmp al, 13
JE CheckNum
mov [si],al
inc si
jmp inputString

CheckNum:
cmp num,'-'
JE PrintNeg
mov dx, offset msgPos
mov ah, 9
int 21h
mov ah,4ch
int 21h

PrintNeg:
mov dx, offset msgNeg
mov ah, 9
int 21h
mov ah,4ch
int 21h
main endp
end main
```