String Compare:

```
.model small

display macro msg
 LEA dx, msg
 mov AH, 09H
 INT 21H

ENDM
.data
msg1 db 0dh, 0AH "Enter first string $"
msg2 db 0dh, 0AH "Enter 2nd string $"
msg3 db 0dh, 0AH "Length of 1String $"
msg4 db 0dh, 0AH "Length of 2 String $"
msg5 db 0dh, 0AH "Are equal ... $"
msg6 db 0dh, 0AH "Are unequal $"

String1 db 80h dup(?)
String2 db 80h dup(?)

.code
START: mov ax, @data
       mov ds, ax
       display msg1
       mov SI, offset String1
       Call readstr
       mov bl, cl
       display msg2
       mov SI, offset string2
       Call read str
       Push Bx
       push CX
       display msg 3
```

```
mov AL, CL
Call len-des
pop CX
pop BX
Cmp CL, BL
JNE FAIL
mov SI, OFFSET STRING1
mov DI, OFFSET STRING2
Cld
CHK: mov AL, [SI]
mov AL, [DI]
JNE FAIL
INC SI
INC DI
dec CL
JNZ CHK
display msg 5
JMP FINAL
Len-disp Proc Near
XOR AH, AH
ADD AL, 00H
AAM
ADD AX, 3030H
mov BH, AL
mov AH, 02H
INT 21H
Ret
Len-dis endp
readstr proc near
XOR CH, CL
Back: mov AH, 01H
INT 21H
```

```
        CMP AL, 0DH
        JE FINISH
        MOV [SI], AL
        INC SI
        INC CL
        JMP BACK
FINISH: MOV [SI], Byte ptr $
        RET
        Readstr endp
        fail: display msg6
        final: mov AH, 4Ch
               int 21h.

END START.
```

NCR

```
model small
.data
n   dw  4
r   dw  2
ncr dw  0

.code
mov ax, @data
mov ds, ax

mov ax, n
mov dx, r
Call ncrpro
Call disp
jmp final

ncrpro proc near
cmp ax, bx
      je res1
    cmp bx, 0
    je res1
    cmp bx, 1
    je resn
    dec ax
    cmp bx, ax
    je inc
  push ax
  push bx
  Call ncrpro
```

```
pop bx
pop ax

dec bx
push ax
push bx
call ncrpro
pop bx
pop ax
ret

res1 : inc ncr
ret

incr : inc ncr
resn : add ncr, ax
ret
ncrpro endp

disp proc near
mov bx, ncr
add bx, 3030h
mov dl, bh
mov ah, 02h
int 21h
mov dl, bl
mov ah, 02h
int 21h
ret
disp endp

Final : mov ah, 4ch
int 21h
```

Decimal up Counter:

```
.MODEL Small
.code
mov AH, 2CH
INT 21H

mov AL, CH
AAM
mov BX, AX
CALL DISP
mov DL, ;
Mov AH, 02H
INT 21H
mov AL, CL
AAM

mov AL, DH
AAM
mov BX, AX
CALL DISP

Mov AH, 4CH
INT 21H
disp proc near
add all, 30H
mov dl, bl
add dl, 30 h
mov ah, 02h
int 21h
ret disp endp end
```

X-Y - Coordinate Cursor

```
.model small
DISP MACRO MSG
LEA DX, MSG
MOV AH, 09H
INT 21H
END M

.Data
ROW DB 02 DUP (0)
COL DB 02 DUP (0)
MSG 1 DB 0DH, 0AH, "Enter -X - coordinate $"
MSG 2 DB 0AH, 0DH, "Enter Y - coordinate $"
MSG 3 DB 0DH, 0AH, "Cursor displayed at $"

.Code
MOV AX, @Data
MOV DS, AX
DISP MSG1
MOV SI, OFFSET ROW
CALL READ
DISP MSG 2
MOV SI, OFFSET COL
CALL READ
MOV SI, OFFSET ROW
MOV AH, [SI]
INC SI
MOV AL, [SI]
SUB AX, 3030H
AAD
```

```
MOV DS, AX
MOV SI, OFFSET [illegible]
MOV AX, [SI]
INC SI
MOV AL, [SI]
SUB AX, 3030H
AAD
MOV DL, AL
MOV AH, 02
MOV AL, 03H
INT 10H
MOV AH, 02H
INT 21H
JMP FINAL
READ PROC NEAR
MOV CX, 02H
BACK: MOV AH, 01H
INT 21H
MOV [SI], AL
INC SI
DEC CX
JNZ BACK
RET
READ ENDP

FINAL : MOV AH, 01H
INT 21H
MOV AH, 4CH
INT 21H
END.
```