[9H]

* **CURSOUR** :↓

```
.MODEL SMALL
                                        //DISP message
DISP MACRO MSG
     LEA DX, MSG
     MOV AH, 09H
     INT 21H
ENDM

.DATA

MSG1  DB  0DH, 0AH, "Enter X-cordinate$"
MSG2  DB  0DH, 0AH, "Enter Y-coordinate.$"
MSG3  DB
          row db 02 dup(0)
.CODE     col db 02 dup(0)

     MOV AX, @DATA              ;always
     MOV DS, AX

     DISP MSG1
     MOV  SI, OFFSET ROW
     CALL READ
     DISP MSG2
     [illegible]
     MOV  SI, OFFSET COL
     CALL READ

     MOV  SI, OFFSET ROW
     MOV  AH, [SI]
     INC  SI
     MOV  AL, [SI]
```

```
→    12
→  01 02
→  31 32
→  01 02
→    12
```

```
SUB AX, 3030
AAD

MOV DH, AL          ; stored in AL

MOV SI, OFFSET COL
MOV AH, [SI]
INC SI
MOV AL, [SI]
SUB AX, 3030
AAD

MOV DL, AL ;        ; cursour pointe values
                      stored in DH, DL
                      x, y.

MOV AH, 00          ; code for setting up the
                      cursos at location
MOV AL, 03H         ;
INT 10H             ;
MOV AH, 02H         ; defaut of the setting a
                      cursor
INT 21H             ;

DISP MSG3
JMP FINAL
```

```
READ PROC NEAR                              ; Two inputs
      MOV CX, 02H                             to be taken

BACK : MOV AH, 01H                          ; interrupt for
       INT 21H                              ; input acceptance
       MOV [SI], AL                         ; Keyboard entry from AL
       INC SI
       DEC CX
   JNZ BACK
   RET

READ ENDP                                   ; for all procedures ENDP

FINAL : MOV AH, 01H                         ; if enter pressed
        INT 21H                               the program terminates
        MOV AH, 4CH                         ; normal
        INT 21H                               end interuppt.
        END
```

we can also write only ENDP