

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                      : pe16.asm
4:  ; FUNCION                       : Programa que compara dos numeros.
5:  ; REALIZADO POR                 : Prof. Juan Ju?rez Fuentes
6:  ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7:  ; FECHA                        : 20240502090049
8:  ;=====
9:  ; DECLARACION DE CONSTANTES
10: ;=====
11: CR          EQU 13      ; Constante CR (Retorno de carro)
12: LF          EQU 10      ; Constante LF (Salto de l?nea)
13: TECLADO_8   EQU 8
14: IMPRIMIR_9  EQU 9
15: FIN_4C00H   EQU 4C00H
16: DOS_21H     EQU 21H
17: BIOS_10H    EQU 10H
18: TAB         EQU 09H
19: BELL        EQU 07H
20: ;=====
21: ; DECLARACION DEL SEGMENTO DE DATOS
22: ;=====
23: DATOS SEGMENT
24:     RENGLOON DB 2
25:     COLUMNA  DB 8
26:     ;-----
27:     MENSAJE  DB 'PROGRAMA QUE MUESTRA SI UN NUMERO ES MAYOR, MENOR O IGUAL QUE
                OTRO.',CR,LF,'$'
28:     INPUT    DB 10,13,'  Ingresas un numero: $'
29:     letrero1 DB 10,13,'  El segundo numero es MAYOR $'
30:     letrero2 DB 10,10,13,13,'  El segundo numero es MENOR.$'
31:     letrero3 DB 10,10,13,13,'  Los numeros son IGUALES.$'
32:     n1       DB ?
33:     n2       DB ?
34: DATOS ENDS
35: ;=====
36: ; DECLARACION DEL SEGMENTO DE DATOS
37: ;=====
38: PILA SEGMENT STACK
39:     db 32 DUP('0')
40: PILA ENDS
41: ;=====
42: ; DECLARACION DEL SEGMENTO DE CODIGO
43: ;=====
44: CODIGO SEGMENT
45:     ;-----
46:     MAIN PROC FAR ; Inicio procedimiento MAIN
47:     ;-----
48:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
49:     ;-----
50:     CALL ENCABEZADO
51:     ;-----
52:     MOV AX,DATOS
53:     MOV DS,AX
54:
55:     MOV AH, 9      ; SERVICIO DE IMPRESION
56:     LEA DX, INPUT  ; OBTIENE LA DIRECCION DE M1
57:     INT 21H        ; INTERRUPCION 21H
58:
59:     MOV AH,01h     ; solicito al usuario que digite un numero
60:     INT 21h
61:     SUB AL,30h     ; realizo el ajuste
62:     MOV n1,AL
63:
64:     MOV AH, 9      ; SERVICIO DE IMPRESION

```

```

65:         LEA DX, INPUT ; OBTIENE LA DIRECCION DE M1
66:         INT 21H      ; INTERRUPCION 21H
67:
68:         MOV AH,01h    ; solicito al usuario que digite el numero dos
69:         INT 21h
70:         SUB AL,30h     ; realizo el ajuste
71:         MOV n2,AL
72:
73:         MOV BL,n1      ; guardo en los registros los numeros que tenia definidos
74:         MOV BH,n2
75:
76:         CMP BH,BL      ; los comparo, como es BH con BL, si bh es mayor ve a la
                        ; etiqueta mayor
77:         JA mayor
78:         JB menor      ; si BH es menor a BL ve a la etiqueta menor
79:         JE iguales    ; si son iguales ve a la etiqueta iguales
80:
81:     menor:
82:         MOV AH,09h     ; imprime es menor
83:         LEA DX,letrero2
84:         INT 21h
85:         JMP salida    ; salto incondicional a la salida
86:     mayor:
87:         MOV AH,09h     ; imprime es mayor
88:         LEA DX,letrero1
89:         INT 21h
90:         JMP salida    ; salta a la salida
91:     iguales:
92:         MOV AH,09h     ; imprime son iguales
93:         LEA DX,letrero3
94:         INT 21h
95:         JMP salida
96:
97:     salida:
98:         ;-----
99:         CALL FIN
100:        ;-----
101:    MAIN ENDP          ; Fin del procedimiento MAIN
102:        ;-----
103:    IMPRIME_TEXTO PROC NEAR
104:        MOV AH,IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
105:        INT DOS_21H      ; Llamar a la interrupci?n del DOS
106:        RET
107:    IMPRIME_TEXTO ENDP
108:        ;-----
109:    FIN PROC NEAR
110:        MOV AX,FIN_4C00H ; Fin del programa
111:        INT DOS_21H
112:        RET
113:    FIN ENDP
114:        ;-----
115:    LIMPIA_PANTALLA PROC NEAR
116:        MOV AX,0600H     ; Establecer modo texto y CLS
117:        MOV BH,1FH       ; Colores de fondo (negro=0) y texto (verde=A).
118:        MOV CX,0000H     ; Desde el inicio de pantalla
119:        MOV DX,484FH     ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
120:        INT BIOS_10H     ; Interrupci?n que llama al BIOS
121:        RET
122:    LIMPIA_PANTALLA ENDP
123:        ;-----
124:    POSICIONA_CURSOR PROC NEAR
125:        MOV AH,02        ; Servicio para Fija posicion del Cursor
126:        MOV BH,00        ; Numero de pagina o pantalla
127:        MOV DH,REGLON
128:        MOV DL,COLUMNA

```

```
129:         INT BIOS_10H
130:         RET
131:     POSICIONA_CURSOR ENDP
132:     ;-----
133:     ENCABEZADO PROC NEAR
134:         CALL LIMPIA_PANTALLA
135:         ;-----
136:         MOV AX,DATOS ; AX=Direcci?n del segmento de datos
137:         MOV DS,AX    ; DS=AX. Indicar el segmento de datos
138:         ;-----
139:         CALL POSICIONA_CURSOR ; Llama a procedimiento par posionar cursor.
140:         ;-----
141:         LEA DX,MENSAJE ; DS:DX = direcci?n de MENSAJE
142:         CALL IMPRIME_TEXTO
143:         ;-----
144:         RET
145:     ENCABEZADO ENDP
146:     ;=====
147: CODIGO ENDS
148:     ;=====
149: END MAIN
150:     ;=====
151:     ;=====
152:     ;=====
```