

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                      : pe12.asm
4:  ; FUNCION                        : Programa que muestra un triangulo de
    asteriscos.
5:  ; REALIZADO POR                  : Prof. Juan Ju?rez Fuentes
6:  ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7:  ; FECHA                          : 20240417125051
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:     RENGLON   DB 2
25:     COLUMNA   DB 18
26: ;-----
27:     MENSAJE   DB 'PROGRAMA QUE MUESTRA UN TRIANGULO DE ASTERISCOS',CR,LF,'$'
28:     ASTERISCO DB '*', '$'
29:     X         DB 28h
30:     Y         DB 5h
31:     AUX       DB 3 ; CANTIDAD DE RENGLONES
32:     CANTIDAD  DB 01h
33: DATOS ENDS
34: ;=====
35: ; DECLARACION DEL SEGMENTO DE DATOS
36: ;=====
37: PILA SEGMENT STACK
38:     db 32 DUP('0')
39: PILA ENDS
40: ;=====
41: ; DECLARACION DEL SEGMENTO DE CODIGO
42: ;=====
43: CODIGO SEGMENT
44: ;-----
45:     MAIN PROC FAR ; Inicio procedimiento MAIN
46: ;-----
47:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
48: ;-----
49:     CALL INTERFAZ
50: ;-----
51:     MOV AX, DATOS
52:     MOV DS, AX
53:
54:     XOR CX, CX ;Reseteo del contador principal (tambien se puede usar mov cx
    ,0000h
55:
56:     primerC:
57:         MOV AH, 02h ; Aca posicionamos el cursor en una parte de la pantalla
58:         MOV BH, 00h
59:         MOV DH, Y    ; coordenadas de las filas
60:         MOV DL, X    ; coordenadas de las columnas
61:         INT 10h
62:
63:         MOV BX, CX ; Almaceno temporalmente el contador principal

```

```

64:
65:     XOR CX, CX ; Reseteo del contador
66:     segundoC: ; Ciclo para el pintado de los asteriscos
67:     MOV AH, 02h ; Aca imprimimos el * el numero de veces que se le indique
68:     MOV DL, ASTERISCO
69:     INT 21h
70:
71:     INC CL
72:     CMP CL, CANTIDAD ; compara si la cantidad de * es igual a la cantidad que
                        ; se debe pintar
73:     JB segundoC ; si es menor continuara pintando
74:
75:     MOV CX, BX ; Recupero el contador principal
76:
77:     DEC X ; aca actualizamos la columna donde se va a posicionar ahora
                        ; el cursor
78:     INC Y ; aca actualizamos la fila donde se va a posicionar ahora el
                        ; cursor
79:
80:     ADD CANTIDAD, 2 ; le aumentamos dos a la cantidad de * que debe pintar
81:
82:     INC CL ; incrementa en 1 el contador cl
83:
84:     CMP CL, AUX ; Repetir n veces
85:     JB primerC
86:
87:     ; -----
88:     CALL FIN
89:     ; -----
90: MAIN ENDP ; Fin del procedimiento MAIN
91:     ; -----
92: IMPRIME_TEXTO PROC NEAR
93:     MOV AH, IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
94:     INT DOS_21H ; Llamar a la interrupci?n del DOS
95:     RET
96: IMPRIME_TEXTO ENDP
97:     ; -----
98: FIN PROC NEAR
99:     MOV AX, FIN_4C00H ; Fin del programa
100:    INT DOS_21H
101:    RET
102: FIN ENDP
103:    ; -----
104: LIMPIA_PANTALLA PROC NEAR
105:    MOV AX, 0600H ; Establecer modo texto y CLS
106:    MOV BH, 1FH ; Colores de fondo (negro=0) y texto (verde=A).
107:    MOV CX, 0000H ; Desde el inicio de pantalla
108:    MOV DX, 484FH ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
109:    INT BIOS_10H ; Interrupci?n que llama al BIOS
110:    RET
111: LIMPIA_PANTALLA ENDP
112:    ; -----
113: POSICIONA_CURSOR_1 PROC NEAR
114:    MOV AH, 02 ; Servicio para Fija posicion del Cursor
115:    MOV BH, 00 ; Numero de pagina o pantalla
116:    MOV DH, RENGLON
117:    MOV DL, COLUMNNA
118:    INT BIOS_10H
119:    RET
120: POSICIONA_CURSOR_1 ENDP
121:    ; -----
122: POSICIONA_CURSOR_2 PROC NEAR
123:    MOV AH, 02 ; Servicio para Fija posicion del Cursor
124:    MOV BH, 00 ; Numero de pagina o pantalla
125:    MOV DH, 3

```

```
126:         MOV DL,5
127:         INT BIOS_10H
128:         RET
129:     POSICIONA_CURSOR_2 ENDP
130:     ;-----
131:     INTERFAZ PROC NEAR
132:         CALL LIMPIA_PANTALLA
133:         ;-----
134:         MOV AX,DATOS ; AX=Direcci?n del segmento de datos
135:         MOV DS,AX    ; DS=AX. Indicar el segmento de datos
136:         ;-----
137:         CALL POSICIONA_CURSOR_1 ; Llama a procedimiento par posionar cursor.
138:         ;-----
139:         LEA DX,MENSAJE ; DS:DX = direcci?n de MENSAJE
140:         CALL IMPRIME_TEXTO
141:         ;-----
142:         CALL POSICIONA_CURSOR_2
143:         RET
144:     INTERFAZ ENDP
145:     ;=====
146: CODIGO ENDS
147:     ;=====
148: END MAIN
149:     ;=====
150:     ;=====
151:     ;=====
```