

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                      : pe20.asm
4:  ; FUNCION                       : Programa para Fibonacci.
5:  ; REALIZADO POR                 : Prof. Juan Juárez Fuentes
6:  ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7:  ; FECHA                        : 20240516094519
8:  ;=====
9:  ; DECLARACION DE CONSTANTES
10: ;=====
11: CR          EQU 13      ; Constante CR (Retorno de carro)
12: LF          EQU 10      ; Constante LF (Salto de línea)
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 18
26: ;-----
27:     MENSAJE  DB 'PROGRAMA QUE MUESTRA: LA SERIE DE FIBONACCI.',CR,LF,'$'
28:     contador DW 2
29:     arreglo  DB 13 dup (0)
30:     n3       DB 0
31:
32: DATOS ENDS
33: ;=====
34: ; DECLARACION DEL SEGMENTO DE DATOS
35: ;=====
36: PILA SEGMENT STACK
37:     db 32 DUP('0')
38: PILA ENDS
39: ;=====
40: ; DECLARACION DEL SEGMENTO DE CODIGO
41: ;=====
42: CODIGO SEGMENT
43: ;-----
44:     MAIN PROC FAR ; Inicio procedimiento MAIN
45: ;-----
46:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
47: ;-----
48:     CALL ENCABEZADO
49: ;-----
50:     MOV AX,DATOS
51:     MOV DS,AX
52:
53: ;-----
54: ; Posicionamiento para la serie de Fibonacci.
55: ;-----
56:     MOV AH,02      ; Servicio para Fija posicion del Cursor
57:     MOV BH,00      ; Numero de pagina o pantalla
58:     MOV DH,4
59:     MOV DL,18
60:     INT BIOS_10H
61: ;-----
62:
63:     MOV AH,0
64:     MOV AL,1
65:

```

```
66:         MOV arreglo[0],AH
67:         MOV arreglo[1],AL
68:
69:         MOV SI,0
70:         MOV DI,1
71:
72:         MOV CX,13d
73:
74:         ciclo:
75:         MOV AL,arreglo[SI]
76:         MOV BL,arreglo[DI]
77:         ADD AL,BL
78:         MOV BX,contador
79:         MOV arreglo[BX],AL
80:         INC SI
81:         INC DI
82:         INC contador
83:         LOOP ciclo
84:
85:         MOV SI,0
86:         MOV CX,13d
87:
88:         ciclo_imprime:
89:         MOV AX,0
90:         MOV AL,arreglo[SI]
91:         CMP AX,144d
92:         JGE ajusta_3_digitos ; Salta si es m?s grande o igual o salta si no es
                               menor que.
93:                               ; El salto se realiza si SF = OF
94:         ajustar_2_digitos:
95:         AAM ; Ajuste
96:         MOV BX,AX
97:         ADD BX,3030h
98:
99:         MOV AH,02h
100:        MOV DL,BH
101:        INT 21h
102:
103:        MOV AH,02h
104:        MOV DL,BL
105:        INT 21h
106:
107:        MOV AH,02h
108:        MOV DL,' '
109:        INT 21h
110:        JMP salta
111:
112:        ajusta_3_digitos:
113:        AAM ; Ajuste
114:        ADD AL,30h
115:        MOV n3,AL
116:
117:        MOV AL,AH
118:        MOV AH,0
119:        AAM ; Ajuste
120:        MOV BX,AX
121:        ADD BX,3030h
122:
123:        MOV AH,02h
124:        MOV DL,BH
125:        INT 21h
126:
127:        MOV AH,02h
128:        MOV DL,BL
129:        INT 21h
```

```

130:
131:         MOV AH,02h
132:         MOV DL,n3
133:         INT 21h
134:
135:         MOV AL,02h
136:         MOV DL,' '
137:         INT 21h
138:
139:         salta:
140:         INC SI
141:         LOOP ciclo_imprime
142:
143:         ;-----
144:         CALL FIN
145:         ;-----
146: MAIN ENDP          ; Fin del procedimiento MAIN
147:         ;-----
148: IMPRIME_TEXTO PROC NEAR
149:         MOV AH,IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
150:         INT DOS_21H       ; Llamar a la interrupci?n del DOS
151:         RET
152: IMPRIME_TEXTO ENDP
153:         ;-----
154: FIN PROC NEAR
155:         MOV AX,FIN_4C00H   ; Fin del programa
156:         INT DOS_21H
157:         RET
158: FIN ENDP
159:         ;-----
160: LIMPIA_PANTALLA PROC NEAR
161:         MOV AX,0600H       ; Establecer modo texto y CLS
162:         MOV BH,1FH         ; Colores de fondo (negro=0) y texto (verde=A).
163:         MOV CX,0000H       ; Desde el inicio de pantalla
164:         MOV DX,484FH       ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
165:         INT BIOS_10H       ; Interrupci?n que llama al BIOS
166:         RET
167: LIMPIA_PANTALLA ENDP
168:         ;-----
169: POSICIONA_CURSOR PROC NEAR
170:         MOV AH,02          ; Servicio para Fija posicion del Cursor
171:         MOV BH,00          ; Numero de pagina o pantalla
172:         MOV DH,REGLON
173:         MOV DL,COLUMNA
174:         INT BIOS_10H
175:         RET
176: POSICIONA_CURSOR ENDP
177:         ;-----
178: ENCABEZADO PROC NEAR
179:         CALL LIMPIA_PANTALLA
180:         ;-----
181:         MOV AX,DATOS ; AX=Direcci?n del segmento de datos
182:         MOV DS,AX    ; DS=AX. Indicar el segmento de datos
183:         ;-----
184:         CALL POSICIONA_CURSOR ; Llama a procedimiento par posionar cursor.
185:         ;-----
186:         LEA DX,MENSAJE ; DS:DX = direcci?n de MENSAJE
187:         CALL IMPRIME_TEXTO
188:         ;-----
189:         RET
190: ENCABEZADO ENDP
191: ;=====
192: CODIGO ENDS
193: ;=====
194: END MAIN

```

```
195: ;=====
196: ;=====
197: ;=====
```