

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                : pell.asm
4:  ; FUNCION                  : Programa que muestra un menu.
5:  ; REALIZADO POR            : Prof. Juan Ju?rez Fuentes
6:  ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7:  ; FECHA                    : 20240415114200
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 MENU',CR,LF,'$'
28:     letrero1 db 10,13,'1.Suma $'
29:     letrero2 db 10,13,'2.Resta $'
30:     letrero3 db 10,13,'3.Multiplicacion$'
31:     letrero4 db 10,13,'4.Division$'
32:     letrero5 db 10,13,'5.Cambiar a azul la pantalla$'
33:     letrero6 db 10,13,'6.Salir$'
34:     letrero7 db 10,13,'Selecciona una opcion $'
35:     letrero8 db 10,13,'Ingresa un numero: $'
36:     letrero9 db 10,13,'El resultado es: $'
37:     n1 db ?
38:     n2 db ?
39: DATOS ENDS
40: ;=====
41: ; DECLARACION DEL SEGMENTO DE DATOS
42: ;=====
43: PILA SEGMENT STACK
44:     DW 100 DUP('0')
45: PILA ENDS
46: ;=====
47: ; DECLARACION DEL SEGMENTO DE CODIGO
48: ;=====
49: CODIGO SEGMENT
50:     ;-----
51:     MAIN PROC FAR ; Inicio procedimiento MAIN
52:     ;-----
53:     ASSUME CS:CODIGO,DS:DATOS,SS:PILA
54:     ;-----
55:     CALL INTERFAZ
56:     ;-----
57:     MOSTRAR_MENU:      ; etiqueta donde vamos a redireccionar nuestro menu
58:     MOV AH,09H         ; imprime el letrero 1
59:     LEA DX,letrero1
60:     INT 21H
61:
62:     LEA DX,letrero2
63:     INT 21H
64:
65:     LEA DX,letrero3

```

```
66:         INT 21H
67:
68:         LEA DX,letrero4
69:         INT 21H
70:
71:         LEA DX,letrero5
72:         INT 21H
73:
74:         LEA DX,letrero6
75:         INT 21H
76:
77:         LEA dx,letrero7
78:         INT 21H
79:
80:         MOV AH,01H      ; solicita una opcion al usuario
81:         INT 21H
82:
83:         CMP AL,31H      ; compara las opciones que el usuario ingresa
84:         JE sub_opcion1  ; si presiono 1 realizara un salto intermedio a la opcion
                        1
85:         CMP AL,32H      ; si presiono 2 realiza el salto intermedio a la opcion 2
86:         JE sub_opcion2
87:         CMP AL,33H
88:         JE sub_opcion3
89:         CMP AL,34H
90:         JE sub_opcion4
91:         CMP AL,35H
92:         JE sub_opcion5
93:         CMP AL,36H
94:         JE sub_opcion6
95:
96:         JMP MOSTRAR_MENU ; si no actua por default y va de nuevo a presentar
                        el menu
97:
98: sub_opcion1:
99:         JMP opcion1     ; cuando llegamos a esta opcion saltamos a la primera
                        opcion
100: sub_opcion2:
101:         JMP opcion2
102: sub_opcion3:
103:         JMP opcion3
104: sub_opcion4:
105:         JMP opcion4
106: sub_opcion5:
107:         JMP opcion5
108: sub_opcion6:
109:         JMP opcion6
110:
111: opcion1:      ; opcion 1 para hacer sumas
112:         ;-----
113:         CALL INTERFAZ
114:         ;-----
115:
116:         MOV AH,09H
117:         LEA DX,letrero8
118:         INT 21H
119:
120:         MOV AH,01H
121:         INT 21H
122:
123:         SUB AL,30H
124:         MOV n1,AL
125:         ;-----
126:         MOV AH,09H
127:         LEA DX,letrero8
```

```
128:         INT 21H
129:
130:         MOV AH,01H
131:         INT 21H
132:
133:         SUB AL,30H
134:         MOV n2,AL
135:
136:         ;-----suma-----
137:         MOV BH,n1
138:         MOV BL,n2
139:
140:         ADD BH,BL
141:
142:         MOV AH,09H
143:         LEA DX,letrero9
144:         INT 21H
145:
146:         MOV AH,02H
147:         MOV DL,BH
148:         ADD DL,30H
149:         INT 21H
150:
151:         MOV AH,01H
152:         INT 21H
153:
154:
155:         JMP MOSTRAR_MENU
156:
157: opcion2:
158:         ;-----
159:         CALL INTERFAZ
160:         ;-----
161:         MOV AH,09H
162:         LEA DX,letrero8
163:         INT 21H
164:
165:         MOV AH,01H
166:         INT 21H
167:
168:         SUB AL,30H
169:         MOV n1,AL
170:         ;-----
171:         MOV AH,09H
172:         LEA DX,letrero8
173:         INT 21H
174:
175:         MOV AH,01H
176:         INT 21H
177:
178:         SUB AL,30H
179:         MOV n2,AL
180:
181:         ;-----Resta-----
182:         MOV BH,n1
183:         MOV BL,n2
184:
185:         SUB BH,BL
186:         ;-----Fin Resta-----
187:         MOV AH,09H
188:         LEA DX,letrero9
189:         INT 21H
190:
191:         MOV AH,02H
192:         MOV DL,BH
```

```
193:         ADD DL,30H
194:         INT 21H
195:
196:         MOV AH,01H
197:         INT 21H
198:
199:         JMP MOSTRAR_MENU
200:
201: opcion3:
202:         ;-----
203:         CALL INTERFAZ
204:         ;-----
205:         MOV AH,09H
206:         LEA DX,letrero8
207:         INT 21H
208:
209:         MOV AH,01H
210:         INT 21H
211:
212:         SUB AL,30H
213:         MOV n1,AL
214:         ;-----
215:         MOV AH,09H
216:         LEA DX,letrero8
217:         INT 21H
218:
219:         MOV AH,01H
220:         INT 21H
221:
222:         SUB AL,30H
223:         MOV n2,AL
224:
225:         ;----Mult-----
226:         MOV AL,n1      ; aca por regla los guardamos en estos registros
227:         MOV BL,n2      ; si no ensamblador no nos va a dar el resultado correcto
228:         MUL BL         ; REALIZAMOS LA MULTIPLICACION
229:         MOV CL,AL      ; guardamos el resultado que se guarda en AL
230:
231:         ;-----
232:         MOV AH,09H
233:         LEA DX,letrero9
234:         INT 21H
235:
236:         MOV AH,02H
237:         MOV DL,CL
238:         ADD DL,30H
239:         INT 21H
240:
241:         MOV AH,01H
242:         INT 21H
243:
244:         JMP MOSTRAR_MENU
245:
246: opcion4:
247:         ;-----
248:         CALL INTERFAZ
249:         ;-----
250:         MOV AH,09H
251:         LEA DX,letrero8
252:         INT 21H
253:
254:         MOV AH,01H
255:         INT 21H
256:
257:         SUB AL,30H
```

```

258:         MOV n1,AL
259:         ;-----
260:         MOV AH,09H
261:         LEA DX,letrero8
262:         INT 21H
263:
264:         MOV AH,01H
265:         INT 21H
266:
267:         SUB AL,30H
268:         MOV n2,AL
269:
270:         ;-----Division-----
271:         ;proceso division
272:         MOV AX,0H ; IMPORTANTE SIEMPRE LIMPIAR EL REGISTRO AX ANTES DE HACER LA
                DIVISION
273:         MOV AL,n1 ; guardamos nuestros digitos en los registros AL y BL
274:         MOV BL,n2 ; es importante seguir este orden si no nos va a dar el
                resultado correcto
275:         DIV BL ; REALIZO LA DIVISION
276:         MOV CL,AL ; se guarda el resultado en el registro CL por gusto no por
                norma
277:         ; recordemos que el resultado de la division se guarda en el registro AL y
                el mod
278:         ; en el registro AH
279:         ;-----
280:
281:         MOV AH,09H
282:         LEA DX,letrero9
283:         INT 21H
284:
285:         MOV AH,02H
286:         MOV DL,CL
287:         ADD DL,30H
288:         INT 21H
289:
290:         MOV AH,01H
291:         INT 21H
292:
293:         JMP MOSTRAR_MENU
294:
295: opcion5:
296:         CALL POSICIONA_CURSOR_1
297:         CALL LIMPIA_PANTALLA2
298:         JMP MOSTRAR_MENU
299:
300: opcion6:
301:         MOV AH,4ch
302:         INT 21H
303:         JMP MOSTRAR_MENU
304:
305:         RET
306:         ;-----
307:         CALL FIN
308:         ;-----
309: MAIN ENDP ; Fin del procedimiento MAIN
310:         ;-----
311: IMPRIME_TEXTO PROC NEAR
312:         MOV AH,IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
313:         INT DOS_21H ; Llamar a la interrupci?n del DOS
314:         RET
315: IMPRIME_TEXTO ENDP
316:         ;-----
317: FIN PROC NEAR
318:         MOV AX,FIN_4C00H ; Fin del programa

```

```

319:         INT DOS_21H
320:         RET
321:     FIN ENDP
322:     ;-----
323:     LIMPIA_PANTALLA PROC NEAR
324:         MOV AX,0600H      ; Establecer modo texto y CLS
325:         MOV BH,1FH        ; Colores de fondo (negro=0) y texto (verde=A).
326:         MOV CX,0000H      ; Desde el inicio de pantalla
327:         MOV DX,484FH      ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
328:         INT BIOS_10H      ; Interrupci?n que llama al BIOS
329:         RET
330:     LIMPIA_PANTALLA ENDP
331:     ;-----
332:     LIMPIA_PANTALLA2 PROC NEAR
333:         MOV AX,0600H      ; Establecer modo texto y CLS
334:         MOV BH,0AH        ; Colores de fondo (negro=0) y texto (verde=A).
335:         MOV CX,0000H      ; Desde el inicio de pantalla
336:         MOV DX,484FH      ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
337:         INT BIOS_10H      ; Interrupci?n que llama al BIOS
338:         RET
339:     LIMPIA_PANTALLA2 ENDP
340:     ;-----
341:     POSICIONA_CURSOR_1 PROC NEAR
342:         MOV AH,02          ; Servicio para Fija posicion del Cursor
343:         MOV BH,00          ; Numero de pagina o pantalla
344:         MOV DH,REGLON
345:         MOV DL,COLUMNNA
346:         INT BIOS_10H
347:         RET
348:     POSICIONA_CURSOR_1 ENDP
349:     ;-----
350:     POSICIONA_CURSOR_2 PROC NEAR
351:         MOV AH,02          ; Servicio para Fija posicion del Cursor
352:         MOV BH,00          ; Numero de pagina o pantalla
353:         MOV DH,3
354:         MOV DL,5
355:         INT BIOS_10H
356:         RET
357:     POSICIONA_CURSOR_2 ENDP
358:     ;-----
359:     INTERFAZ PROC NEAR
360:         CALL LIMPIA_PANTALLA
361:         ;-----
362:         MOV AX,DATOS      ; AX=Direcci?n del segmento de datos
363:         MOV DS,AX         ; DS=AX. Indicar el segmento de datos
364:         ;-----
365:         CALL POSICIONA_CURSOR_1 ; Llama a procedimiento par posionar cursor.
366:         ;-----
367:         LEA DX,MENSAJE    ; DS:DX = direcci?n de MENSAJE
368:         CALL IMPRIME_TEXTO
369:         ;-----
370:         CALL POSICIONA_CURSOR_2
371:         RET
372:     INTERFAZ ENDP
373:     ;=====
374:     CODIGO ENDS
375:     ;=====
376:     END MAIN
377:     ;=====
378:     ;=====
379:     ;=====

```