

```

1:  ;=====
2:  ;=====
3:  ; PROGRAMA                      : pe15.asm
4:  ; FUNCION                       : Programa que muestra si un numero es primo.
5:  ; REALIZADO POR                 : Prof. Juan Ju?rez Fuentes
6:  ; COMPILADOR EN EL QUE SE EJECUTO: GUI Turbo Assembler Version 5.1
7:  ; FECHA                        : 20240430084934
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 18
26:     ;-----
27:     MENSAJE  DB 'PROGRAMA QUE MUESTRA SI UN NUMERO ES PRIMO',CR,LF,'$'
28:     letrero1 DB 10,13,'  Ingresa un numero: $'
29:     letrero2 DB 10,10,13,'  El numero es primo.$'
30:     letrero3 DB 10,10,13,'  El numero No es primo.$'
31:     n        DB ?
32:     i        DB 0
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 ENCABEZADO
50:     ;-----
51:     MOV AX,DATOS
52:     MOV DS,AX
53:
54:     MOV AH,09h ; funcion para imprimir en pantalla una cadena
55:     LEA DX,letrero1
56:     INT 21h
57:
58:     MOV AH,01h ; solicito al usuario que digite un numero
59:     INT 21h
60:     SUB AL,30h ; realizo el ajuste
61:
62:     MOV n,AL ; lo guardo en mi variable n
63:     MOV CX,0h ; limpio mi contador
64:     MOV CL,n ; le paso como valor maximo el numero que definio el usuario
                   (for...; i<=n;.... )

```

```

65:      MOV SI,0      ; establezco si como un contador que usare y lo posiciono en
                        0
66:      for:          ; comienzo con mi etiqueta for
67:      INC i          ; incremento i en 1 (for INT i=1;...;i++)
68:      MOV AX,0000h   ; para realizar la division siempre se debe limpiar el
                        registro ax
69:      MOV AL,n       ; establecemos el numero como dividendo
70:      MOV BL,i       ; el valor de i sera el divisor
71:      DIV BL         ; realiza la division n/i
72:
73:      CMP AH,0       ; el modulo se guarda en el registro ah y se compara con 0
                        para ver si fue entera la division
74:      JE es_cero     ; si es cero incrementa el valor de SI en 1
75:      JNE no_cero    ; si no sigue iterando con el loop
76:
77:      es_cero:
78:      INC SI
79:      no_cero:
80:      LOOP for
81:
82:      CMP SI,2       ; finalizando el loop compara si el registro SI es 2 por la
                        analogia de un numero primo
83:      JZ primo       ; si es cierto va a imprimir que es primo
84:      JNZ noprimo    ; si no imprimira que no es primo
85:
86:      primo:
87:      MOV AH,09h
88:      LEA DX,letrero2
89:      INT 21h
90:      JMP salir
91:      noprimo:
92:      MOV AH,09h
93:      LEA DX,letrero3
94:      INT 21h
95:      salir:
96:      ;-----
97:      CALL FIN
98:      ;-----
99:      MAIN ENDP      ; Fin del procedimiento MAIN
100:     ;-----
101:     IMPRIME_TEXTO PROC NEAR
102:         MOV AH,IMPRIMIR_9 ; Funci?n DOS para escribir texto en pantalla
103:         INT DOS_21H       ; Llamar a la interrupci?n del DOS
104:         RET
105:     IMPRIME_TEXTO ENDP
106:     ;-----
107:     FIN PROC NEAR
108:         MOV AX,FIN_4C00H   ; Fin del programa
109:         INT DOS_21H
110:         RET
111:     FIN ENDP
112:     ;-----
113:     LIMPIA_PANTALLA PROC NEAR
114:         MOV AX,0600H       ; Establecer modo texto y CLS
115:         MOV BH,1FH         ; Colores de fondo (negro=0) y texto (verde=A).
116:         MOV CX,0000H       ; Desde el inicio de pantalla
117:         MOV DX,484FH       ; Hasta el fin de la pantalla Reng (48)72 col(79)4F
118:         INT BIOS_10H       ; Interrupci?n que llama al BIOS
119:         RET
120:     LIMPIA_PANTALLA ENDP
121:     ;-----
122:     POSICIONA_CURSOR PROC NEAR
123:         MOV AH,02          ; Servicio para Fija posicion del Cursor
124:         MOV BH,00          ; Numero de pagina o pantalla
125:         MOV DH,REGLON

```

```
126:         MOV DL,COLUMNNA
127:         INT BIOS_10H
128:         RET
129:     POSICIONA_CURSOR ENDP
130:     ;-----
131:     ENCABEZADO 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 ; Llama a procedimiento par posionar cursor.
138:         ;-----
139:         LEA DX,MENSAJE ; DS:DX = direcci?n de MENSAJE
140:         CALL IMPRIME_TEXTO
141:         ;-----
142:         RET
143:     ENCABEZADO ENDP
144:     ;=====
145: CODIGO ENDS
146:     ;=====
147: END MAIN
148:     ;=====
149:     ;=====
150:     ;=====
```