

```

1  #ifndef BYTE
2      #define BYTE unsigned char
3  #endif
4
5  enum types { INVISIBLE, NORMAL, GRUESO };
6
7  // Indica si se ha pulsado alguna tecla
8  int kbhit();
9
10 // Indica la posición x actual del cursor
11 int wherex();
12
13 // Indica la posición y actual del cursor
14 int wherey();
15
16 // Mueve el cursor a la posición según las coordenadas especificadas por los argumentos x e y
17 void gotoxy(int x, int y);
18
19 // Fija el aspecto del cursor, debe admitir tres valores: INVISIBLE, NORMAL y GRUESO
20 void setcursortype(enum types);
21
22 // Fija el modo de video deseado
23 void setvideomode(BYTE modo);
24
25 void ajustar_resolucion(int nuevo_modos);
26
27 // Obtiene el modo de video actual
28 int getvideomode();
29
30 // Modifica el color de primer plano con que se mostrarán los caracteres
31 void textcolor(int color);
32
33 // Modifica el color de fondo con que se mostrarán los caracteres
34 void textbackground();
35
36 // Borra toda la pantalla
37 void clrscr();
38
39 // Borra una línea desde la posición actual del cursor hasta el final de la misma
40 void clreol();
41
42 // Desplaza toda la pantalla una línea hacia arriba
43 void scrollup(int n_lineas, int color, int lsi, int csi, int lid, int cid);
44
45 // Desplaza toda la pantalla una línea hacia abajo
46 void scrolldown(int n_lineas, int color, int lsi, int csi, int lid, int cid);
47
48 // Escribe un caracter en pantalla con el color indicado actualmente
49 void cputchar(const char, int);
50
51 // Obtiene el caracter de teclado y lo muestra en pantalla
52 int getche();
53
54 // Obtiene un caracter desde el teclado
55 int getch();
56
57 // Imprime una cadena de caracteres en pantalla
58 int cputs(const char * cadena);
59
60 // Devuelve el color actual en decimal (mezcla de color de fondo y el color de texto)
61 int getcolor();
62
63 // Avanza el cursor: una columna (si puede) o una línea y reinicia la columna a 0
64 void avanzar_cursor()

```