

```

1  #ifndef NEWGRAPHICS_H
2  #define NEWGRAPHICS_H
3
4  #include<windows.h>
5  #include<dos.h>
6  #include<iostream>
7  #include<stdio.h>
8  HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE); // lay handle cua console hien tai
9
10 /* Ham: gotoxy()
11  * Chuc nang: dua con tro den vi tri x, y
12  * Tham so: x, y
13  * Tra ve: khong
14  */
15 void gotoxy(short x, short y)
16 {
17     HANDLE hConsoleOutput;
18     COORD Cursor_an_Pos = { x,y};
19     hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);
20     SetConsoleCursorPosition(hConsoleOutput , Cursor_an_Pos);
21 }
22
23 /* Ham: SetColor()
24  * Chuc nang: dat mau cho chu
25  * Tham so: color
26  * Tra ve: khong
27  */
28 void SetColor(WORD color)
29 {
30     HANDLE hConsoleOutput;
31     hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);
32
33     CONSOLE_SCREEN_BUFFER_INFO screen_buffer_info;
34     GetConsoleScreenBufferInfo(hConsoleOutput, &screen_buffer_info);
35
36     WORD wAttributes = screen_buffer_info.wAttributes;
37     color &= 0x000f;
38     wAttributes &= 0xffff0;
39     wAttributes |= color;
40
41     SetConsoleTextAttribute(hConsoleOutput, wAttributes);
42 }
43
44 /* Ham: SetBGColor()
45  * Chuc nang: dat mau nen cho chu
46  * Tham so: color
47  * Tra ve: khong
48  */
49 void SetBGColor(WORD color)
50 {
51     HANDLE hConsoleOutput;
52     hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);
53
54     CONSOLE_SCREEN_BUFFER_INFO screen_buffer_info;
55     GetConsoleScreenBufferInfo(hConsoleOutput, &screen_buffer_info);
56
57     WORD wAttributes = screen_buffer_info.wAttributes;
58     color &= 0x000f;
59     color <= 4;
60     wAttributes &= 0xff0f;
61     wAttributes |= color;
62
63     SetConsoleTextAttribute(hConsoleOutput, wAttributes);
64 }
65
66 /* Ham: NT_SetConsoleDisplayMode()
67  * Chuc nang: thiet lap che do hien thi, co full screen hay khong
68  * Tham so: hOutputHandle, dwNewMode
69  * Tra ve: TRUE or FALSE
70  */
71 BOOL NT_SetConsoleDisplayMode(HANDLE hOutputHandle, DWORD dwNewMode)
72 {
73     typedef BOOL (WINAPI *SCDMPProc_t) (HANDLE, DWORD, LPDWORD);
74     SCDMPProc_t SetConsoleDisplayMode;
75     HMODULE hKernel32;
76     BOOL bFreeLib = FALSE, ret;
77     const char KERNEL32_NAME[] = "kernel32.dll";

```

```

78
79     hKernel32 = GetModuleHandleA(KERNEL32_NAME);
80     if (hKernel32 == NULL)
81     {
82         hKernel32 = LoadLibraryA(KERNEL32_NAME);
83         if (hKernel32 == NULL)
84             return FALSE;
85
86         bFreeLib = TRUE;
87     }
88
89     SetConsoleDisplayMode =
90         (SCDMPProc_t)GetProcAddress(hKernel32, "SetConsoleDisplayMode");
91     if (SetConsoleDisplayMode == NULL)
92     {
93         SetLastError(ERROR_CALL_NOT_IMPLEMENTED);
94         ret = FALSE;
95     }
96     else
97     {
98         DWORD tmp;
99         ret = SetConsoleDisplayMode(hOutputHandle, dwNewMode, &tmp);
100     }
101
102     if (bFreeLib)
103         FreeLibrary(hKernel32);
104
105     return ret;
106 }
107
108 /* Ham: setFullScreen()
109  * Chuc nang: dat che do full screen
110  * Tham so: khong
111  * Tra ve: khong
112  */
113 void setFullScreen()
114 {
115     NT_SetConsoleDisplayMode( GetStdHandle( STD_OUTPUT_HANDLE ), 1 );
116 }
117
118 /* Ham: exitFullScreen()
119  * Chuc nang: thoat khoi full screen
120  * Tham so: khong
121  * Tra ve: khong
122  */
123 void exitFullScreen()
124 {
125     NT_SetConsoleDisplayMode( GetStdHandle( STD_OUTPUT_HANDLE ), 0 );
126 }
127
128 /* Ham: ShowCur()
129  * Chuc nang: an con tro trong console
130  * Tham so: CursorVisibility
131  * Tra ve: khong
132  */
133 void ShowCur(bool CursorVisibility)
134 {
135     HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
136     CONSOLE_CURSOR_INFO cursor = {1, CursorVisibility};
137     SetConsoleCursorInfo(handle, &cursor);
138 }
139
140 /* Ham: cls()
141  * Chuc nang: xoa man hinh console
142  * Tham so: hConsole
143  * Tra ve: khong
144  */
145 void cls( HANDLE hConsole )
146 {
147     COORD coordScreen = { 0, 0 };    // home for the cursor
148     DWORD cCharsWritten;
149     CONSOLE_SCREEN_BUFFER_INFO csbi;
150     DWORD dwConSize;
151
152     // Get the number of character cells in the current buffer.
153
154     if( !GetConsoleScreenBufferInfo( hConsole, &csbi ) )

```

```

155         return;
156         dwConSize = csbi.dwSize.X * csbi.dwSize.Y;
157
158         // Fill the entire screen with blanks.
159
160         if( !FillConsoleOutputCharacter( hConsole, (TCHAR) ' ',
161             dwConSize, coordScreen, &cCharsWritten ) )
162             return;
163
164         // Get the current text attribute.
165
166         if( !GetConsoleScreenBufferInfo( hConsole, &csbi ) )
167             return;
168
169         // Set the buffer's attributes accordingly.
170
171         if( !FillConsoleOutputAttribute( hConsole, csbi.wAttributes,
172             dwConSize, coordScreen, &cCharsWritten ) )
173             return;
174
175         // Put the cursor at its home coordinates.
176
177         SetConsoleCursorPosition( hConsole, coordScreen );
178     }
179
180     /* Ham: clearScreen()
181     * Chuc nang: xoa man hinh console hien tai
182     * Tham so: khong
183     * Tra ve: khong
184     */
185     void clearScreen(){
186         cls(hCon);
187     }
188 #endif // NEWGRAPHICS_H
189
190
191

```