初学编程的读者，肯定很想编写出一个游戏来秀一下，贪吃蛇就是C语言小游戏中经典的一个。  
  
编写游戏的第一步就是搞定光标定位问题，这样你的程序就可以随心所欲、按任意顺序、在任意位置输出了。  
  
光标定位需要使用windows.h头文件中的SetConsoleCursorPosition函数：

SetConsoleCursorPosition(HANDLE hConsoleOutput, COORD dwCursorPosition);

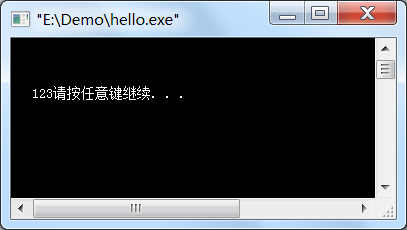
hConsoleOutput表示控制台缓冲区句柄，可通过GetStdHandle(STD\_OUTPUT\_HANDLE)来获得；dwCursorPosition是光标位置，也就是第几行第几列，它是COORD类型的结构体。  
  
例如，将光标定位到第3行第3列：

1. //定义光标位置
2. COORD coord;
3. coord.X = 3; //第3行
4. coord.Y = 3; //第3列
5. //获取控制台缓冲区句柄
6. HANDLE ConsoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);
7. //设置光标位置
8. SetConsoleCursorPosition(ConsoleHandle, coord);

这样，程序就可以从第3行第3列开始输出了。  
  
将代码补充完整：

1. #include <stdio.h>
2. #include <windows.h>
3. int main(){
4. //定义光标位置
5. COORD coord;
6. coord.X = 3; //第3行
7. coord.Y = 3; //第3列
8. //获取控制台缓冲区句柄
9. HANDLE ConsoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);
10. //设置光标位置
11. SetConsoleCursorPosition(ConsoleHandle, coord);
13. printf("123");
14. **return** 0;
15. }

运行结果：

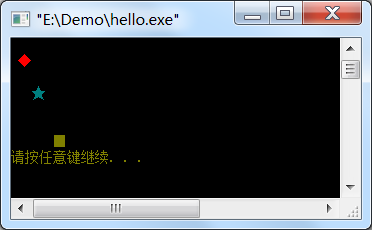


注意：窗口的左上角是第0行第0列。

搞定了光标重定位，你就可以将输出顺序打乱了，想在哪里输出就在哪里输出。例如：

1. #include <stdio.h>
2. #include <windows.h>
3. //设置光标位置
4. void setCursorPosition(int x, int y);
5. //设置文字颜色
6. void setColor(int color);
7. int main(){
8. setColor(3);
9. setCursorPosition(3, 3);
10. puts("★");
11. setColor(0XC);
12. setCursorPosition(1, 1);
13. puts("◆");
14. setColor(6);
15. setCursorPosition(6, 6);
16. puts("■");
17. **return** 0;
18. }
19. void setCursorPosition(int x, int y){
20. COORD coord;
21. coord.X = x;
22. coord.Y = y;
23. SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);
24. }
25. void setColor(int color){
26. SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color) ;
27. }

运行结果：



可以发现，先输出的字符位置不一定靠前，可以在任意位置。

看到这里，很多初学者会不懂，甚至一头雾水，不过没关系，这里仅仅是演示，引起你对C语言的兴趣，待学完C语言的基本语法后会恍然大悟。