```c
#include<windows.h>
#include<stdio.h>       // for sprintf()
#include<process.h> // for _beginthread()

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

// ThreadProc() functions
void __cdecl MyThreadProcOne(void *);
void __cdecl MyThreadProcTwo(void *);

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,
                   LPSTR lpCmdLine,int nCmdShow)
{
    WNDCLASSEX wndclass;
    HWND hwnd;
    MSG msg;
    char AppName[] = "MULTITHREADING";

    wndclass.cbSize = sizeof(wndclass);
    wndclass.style = CS_HREDRAW | CS_VREDRAW;
    wndclass.cbClsExtra = 0;
    wndclass.cbWndExtra = 0;
    wndclass.lpfnWndProc = WndProc;
    wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
    wndclass.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH);
    wndclass.hInstance = hInstance;
    wndclass.lpszClassName = AppName;
    wndclass.lpszMenuName = NULL;

    RegisterClassEx(&wndclass);

    hwnd=CreateWindow(AppName,
                      "Example Of Multithreading",
                WS_OVERLAPPEDWINDOW,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                CW_USEDEFAULT,
                        NULL,
                        NULL,
                hInstance,
                NULL);

    ShowWindow(hwnd,nCmdShow);
    UpdateWindow(hwnd);

    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
```

```
53
54      //Previously for Visual Studio6
55  //  return (msg.wParam);
56      return ((int)msg.wParam);
57  }
58
59  // Window Procedure
60  LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
61  {
62      unsigned long ulThread1,ulThread2;
63
64      switch (iMsg) {
65          case WM_CREATE:
66              //Previously for Visual Studio6
67              //ulThread1=_beginthread(MyThreadProcOne,0,(void *)hwnd);
68              ulThread1=(unsigned long )_beginthread(MyThreadProcOne,0,(void *)
                  hwnd);
69
70              //Previously for Visual Studio6
71              //ulThread2=_beginthread(MyThreadProcTwo,0,(void *)hwnd);
72              ulThread2=(unsigned long )_beginthread(MyThreadProcTwo,0,(void *)
                  hwnd);
73              break;
74
75          case WM_DESTROY:
76              PostQuitMessage(0);
77              break;
78      }
79      return (DefWindowProc(hwnd, iMsg, wParam, lParam));
80  }
81
82  void __cdecl MyThreadProcOne(void *param)
83  {
84      HDC hdc;
85      int i;
86      char str[255];
87
88      hdc = GetDC((HWND)param);
89      for(i = 0; i <=32767; i++) {
90          //Previously for Visual Studio6
91          //sprintf(str, "Thread 1 -> Increasing Order Output = %d", i);
92          sprintf_s(str,255,"Thread 1 -> Increasing Order Output = %d", i);
93          //Previously for Visual Studio6
94          //TextOut(hdc, 5, 5, str, strlen(str));
95          TextOut(hdc, 5, 5, str, (int)strlen(str));
96      }
97      ReleaseDC((HWND)param, hdc);
98  }
99
100 void __cdecl MyThreadProcTwo(void *param)
101 {
102     HDC hdc;
```

```
103     int i;
104     char str[255];
105
106     hdc = GetDC((HWND)param);
107     for(i = 32767; i >= 0; i--) {
108         //Previously for Visual Studio6
109         //sprintf(str, "Thread 2 -> Decreasing Order Output = %d", i);
110         sprintf_s(str,255,"Thread 2 -> Increasing Order Output = %d", i);
111         //Previously for Visual Studio6
112         //TextOut(hdc, 5, 25, str, strlen(str));
113         TextOut(hdc, 5, 25, str, (int)strlen(str));
114     }
115     ReleaseDC((HWND)param,hdc);
116 }
117
```