```
//Programarea in windows. Lab 1.

//Dorian Gotonoaga. IA-181

#include <iostream>
#include <Windows.h>

void MyMenu(HWND hwnd) {
        HMENU hMenu = CreateMenu();
        AppendMenu(hMenu, MF_POPUP, 2, L"LAB 1");
        SetMenu(hwnd, hMenu);
}

HWND hStaticText;
HRGN hrgn;

LRESULT WINAPI WndProc(HWND hwnd, UINT msg, WPARAM wparam, LPARAM lparam) {
        switch (msg)
        {
        case WM_CREATE: {
                RECT rect;
                GetWindowRect(hwnd, &rect);

                HWND hBtn = CreateWindow(
                        L"Button",
                        L"whis btn",
                        WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
                        0,0, 150, 35,
                        hwnd, reinterpret_cast<HMENU>(1), NULL, NULL
                );
                hStaticText = CreateWindow(
                        L"Static",
                        L"Static Centered Text",
                        WS_CHILD | WS_VISIBLE,
                        (rect.right - rect.left) / 2, (rect.bottom - rect.top) / 2, 150, 20,
                        hwnd, NULL, NULL, NULL
                );
                MyMenu(hwnd);
                break;
        }
        case WM_SIZE: {
                RECT rect;
                GetWindowRect(hwnd, &rect);
                SetWindowPos(hStaticText, 0, (rect.right - rect.left - 150) / 2, (rect.bottom - rect.top -
60) / 2, 150, 20, SWP_NOSIZE | SWP_NOZORDER);

                break;
        }
        case WM_PAINT: {
                //RECT rect;
                //GetWindowRect(hwnd, &rect);
                //SetWindowPos(hStaticText, 0, (rect.right - rect.left) / 2, (rect.bottom - rect.top) / 2,
150, 20, SWP_NOSIZE | SWP_NOZORDER);

                break;
        }
        case WM_DESTROY: {
                PostQuitMessage(0);
                break;
        }
        case WM_COMMAND: {
                switch (LOWORD(wparam))
                {
                case 1: {
                        //Button handler
                        break;
                }
                case 2: {
                        DestroyWindow(hwnd);
                        break;
                }
                default:
                        break;
                }
        }
        default:
                return DefWindowProcW(hwnd, msg, wparam, lparam);
                break;
        }
}

int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {
        HWND hwnd;
        MSG msg;
        WNDCLASSEX wc;
        wc.style = CS_HREDRAW | CS_VREDRAW;
        wc.cbSize = sizeof(WNDCLASSEX);
        wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
        wc.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
        wc.hCursor = LoadIcon(NULL, IDC_HAND);
        wc.hbrBackground = (HBRUSH)GetStockObject(GRAY_BRUSH);
        wc.hInstance = hinstance;
        wc.cbClsExtra = 0;
        wc.cbWndExtra = 0;
        wc.lpfnWndProc = WndProc;
        wc.lpszMenuName = NULL;
        wc.lpszClassName = L"WinMainClass";

        RegisterClassEx(&wc);

        hwnd = CreateWindow(
                wc.lpszClassName,
                L"My First App",
                WS_OVERLAPPEDWINDOW,
                0, 0, 500, 400,
                NULL, NULL, hinstance, NULL
        );

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

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

        return msg.wParam;

}
```