**PRACTICAL 2**

**Name:** Smit M Khobragade

**Sec:** A

**Roll no.:** 64

**Aim:** Study of creation of Window.

**Code & Output:**

1. Try to display a window with different combinations of window styles and observe the results.

#include <windows.h>

int \_stdcall WinMain(HINSTANCE *hinstance*, HINSTANCE *hPrevinstance*, LPSTR *IpszCmdline*, int *nCmdShow*)

{

  HWND h[5];

  h[0] = CreateWindow("BUTTON", "HSCROLL", WS\_HSCROLL, 10, 10, 150, 100, 0, 0, hinstance, 0);

  ShowWindow(h[0], nCmdShow);

  h[1] = CreateWindow("BUTTON", "MAXIMIZEBOX", WS\_OVERLAPPEDWINDOW | WS\_MAXIMIZEBOX, 100, 100, 150, 100, 0, 0, hinstance, 0);

  ShowWindow(h[1], nCmdShow);

  h[2] = CreateWindow("BUTTON", "MINIMIZEBOX", WS\_OVERLAPPEDWINDOW | WS\_MINIMIZEBOX, 200, 200, 150, 100, 0, 0, hinstance, 0);

  ShowWindow(h[2], nCmdShow);

  h[3] = CreateWindow("BUTTON", "VSCROLL", WS\_VSCROLL, 300, 300, 150, 100, 0, 0, hinstance, 0);

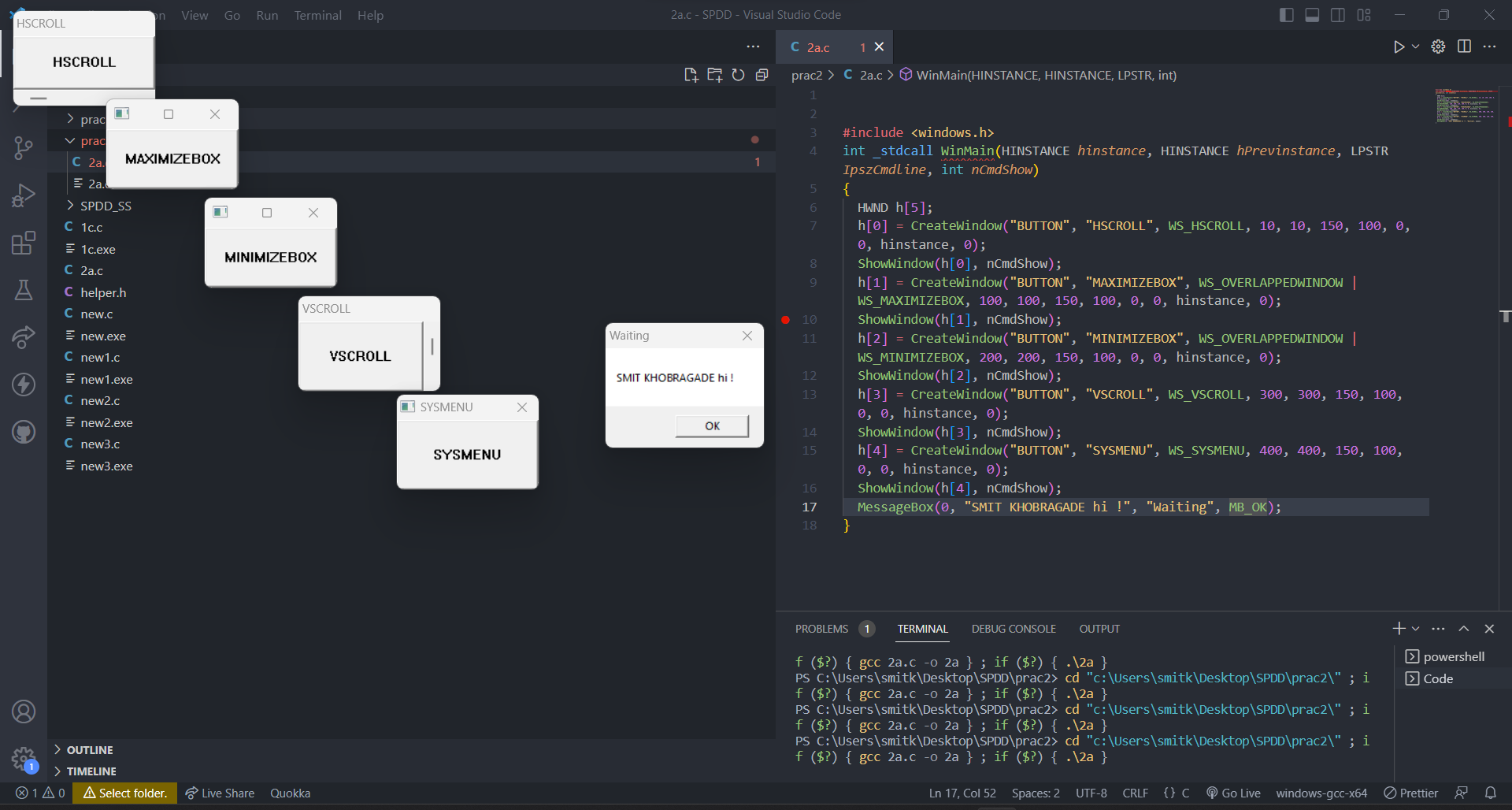
  ShowWindow(h[3], nCmdShow);

  h[4] = CreateWindow("BUTTON", "SYSMENU", WS\_SYSMENU, 400, 400, 150, 100, 0, 0, hinstance, 0);

  ShowWindow(h[4], nCmdShow);

  MessageBox(0, "SMIT KHOBRAGADE hi !", "Waiting", MB\_OK);

}



1. Write a program, which receives an integer as a command line argument, creates an edit window, and based on the value of the integer displays a button window as maximized/minimized / normal.

#include <windows.h>

int \_stdcall WinMain(HINSTANCE *hinstance*, HINSTANCE *hPrevinstance*, LPSTR *lpszCmdline*, int *nCmdShow*)

{

  HWND h;

  int num;

  num = atoi(lpszCmdline);

  h = CreateWindow("BUTTON", "Hit ME", WS\_OVERLAPPEDWINDOW, 10, 10, 150, 100, 0, 0, hinstance, 0);

  switch (num)

  {

  case 0:

    ShowWindow(h, SW\_HIDE);

    break;

  case 1:

    ShowWindow(h, SW\_NORMAL);

    break;

  case 2:

    ShowWindow(h, SW\_MINIMIZE);

    break;

  case 3:

    ShowWindow(h, SW\_MAXIMIZE);

    break;

  default:

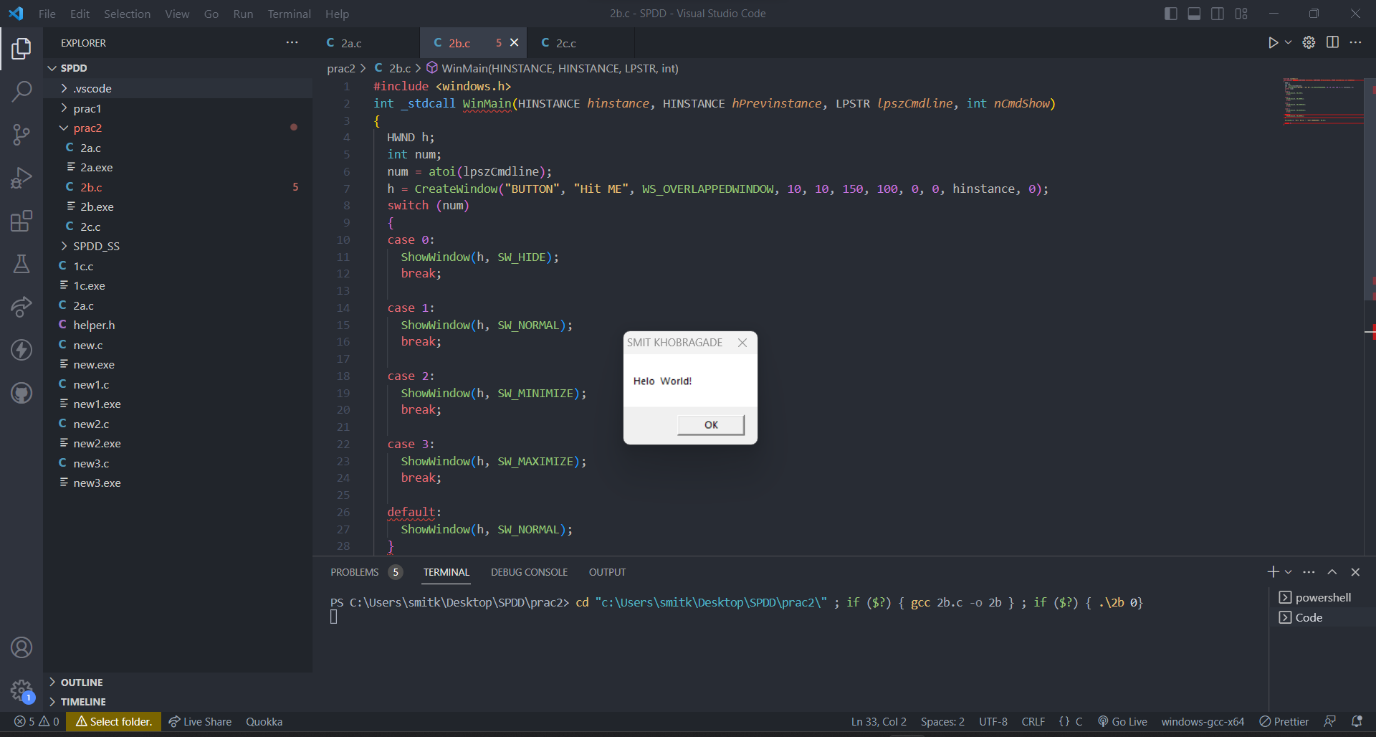
    ShowWindow(h, SW\_NORMAL);

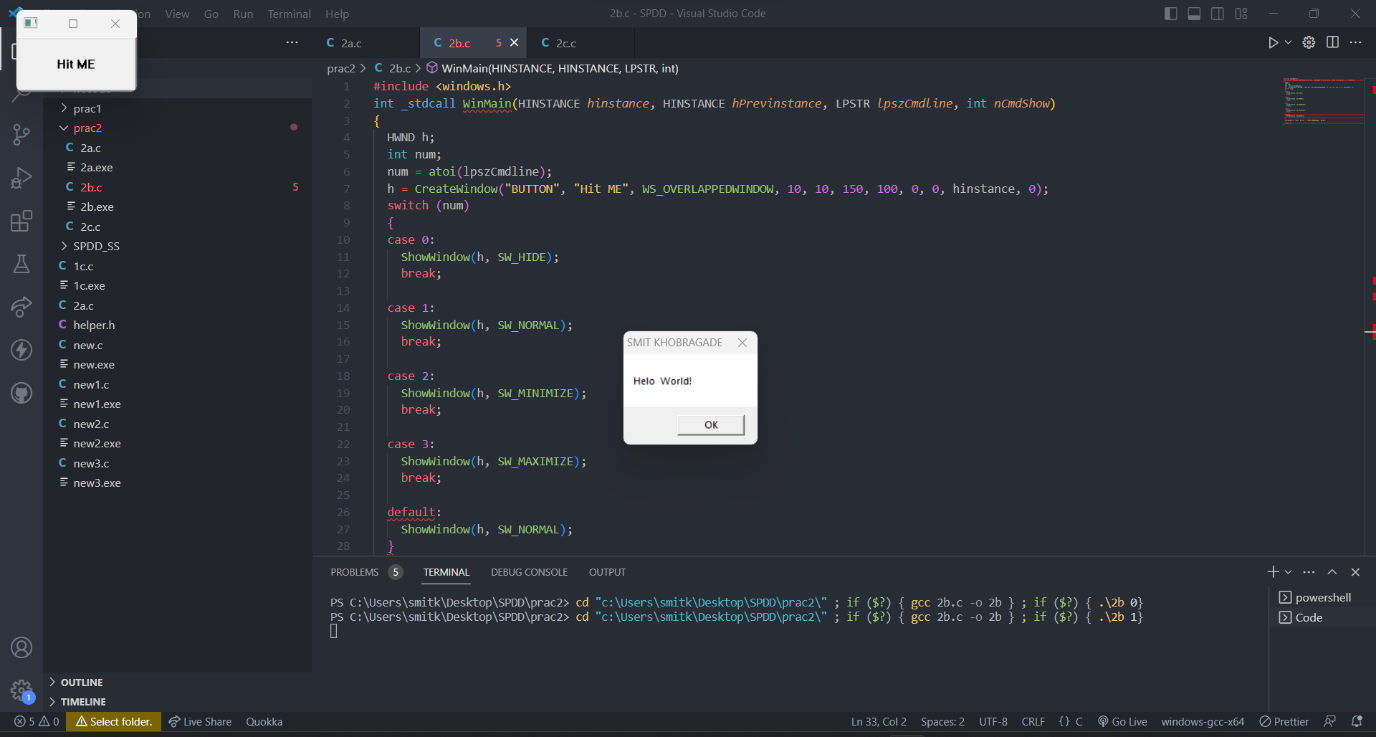
  }

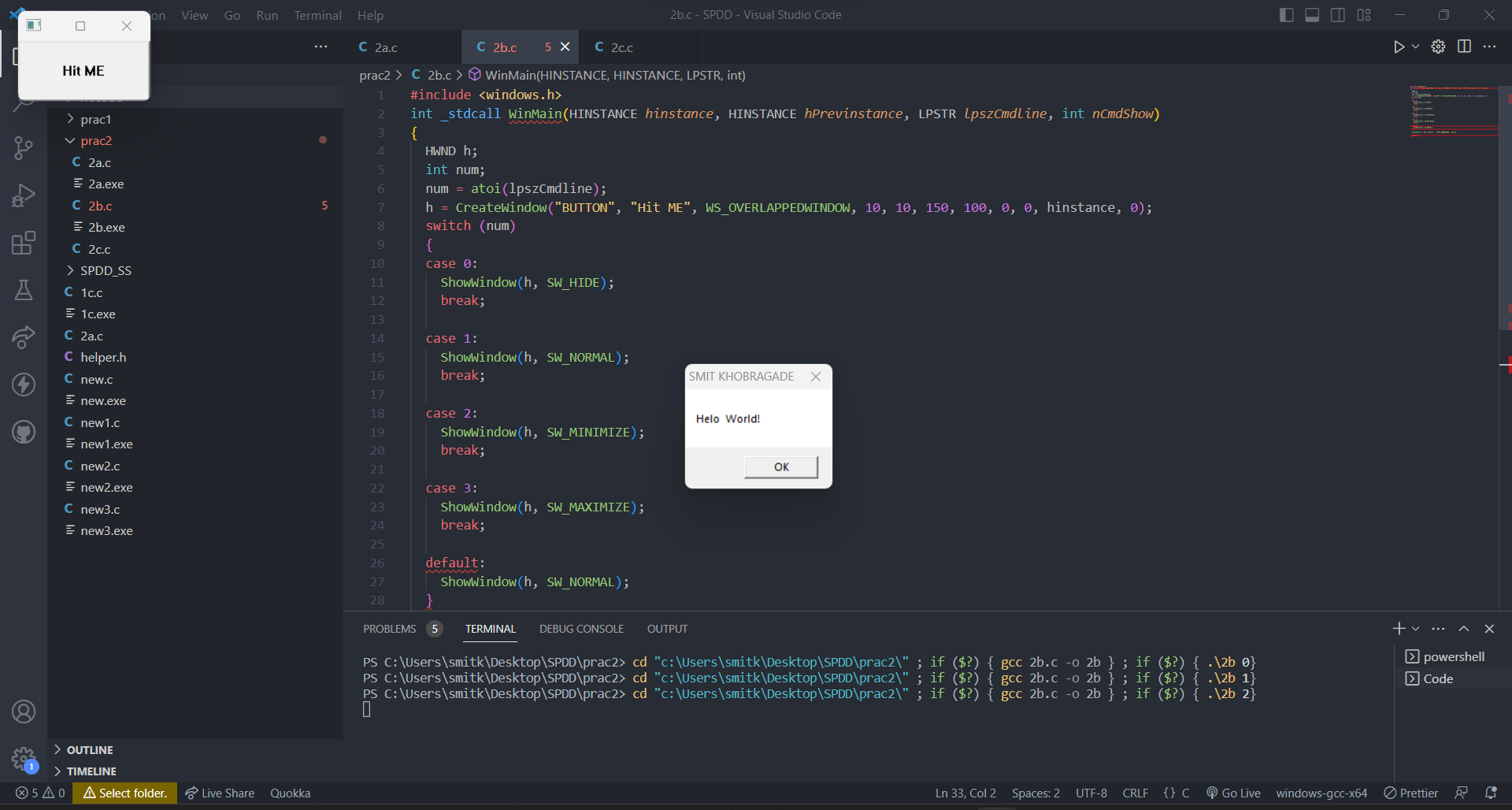
  MessageBox(0, "Helo  World! ", "SMIT KHOBRAGADE", MB\_OK);

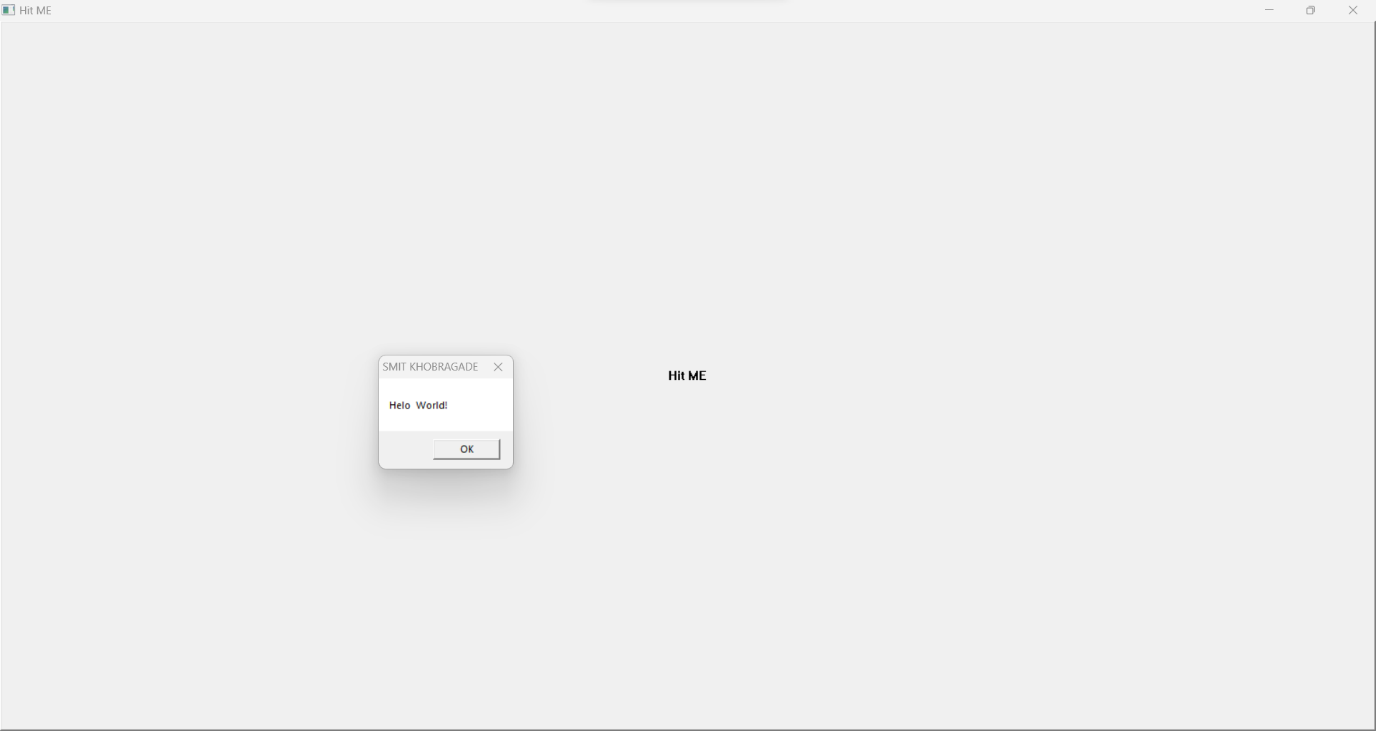
  return 0;

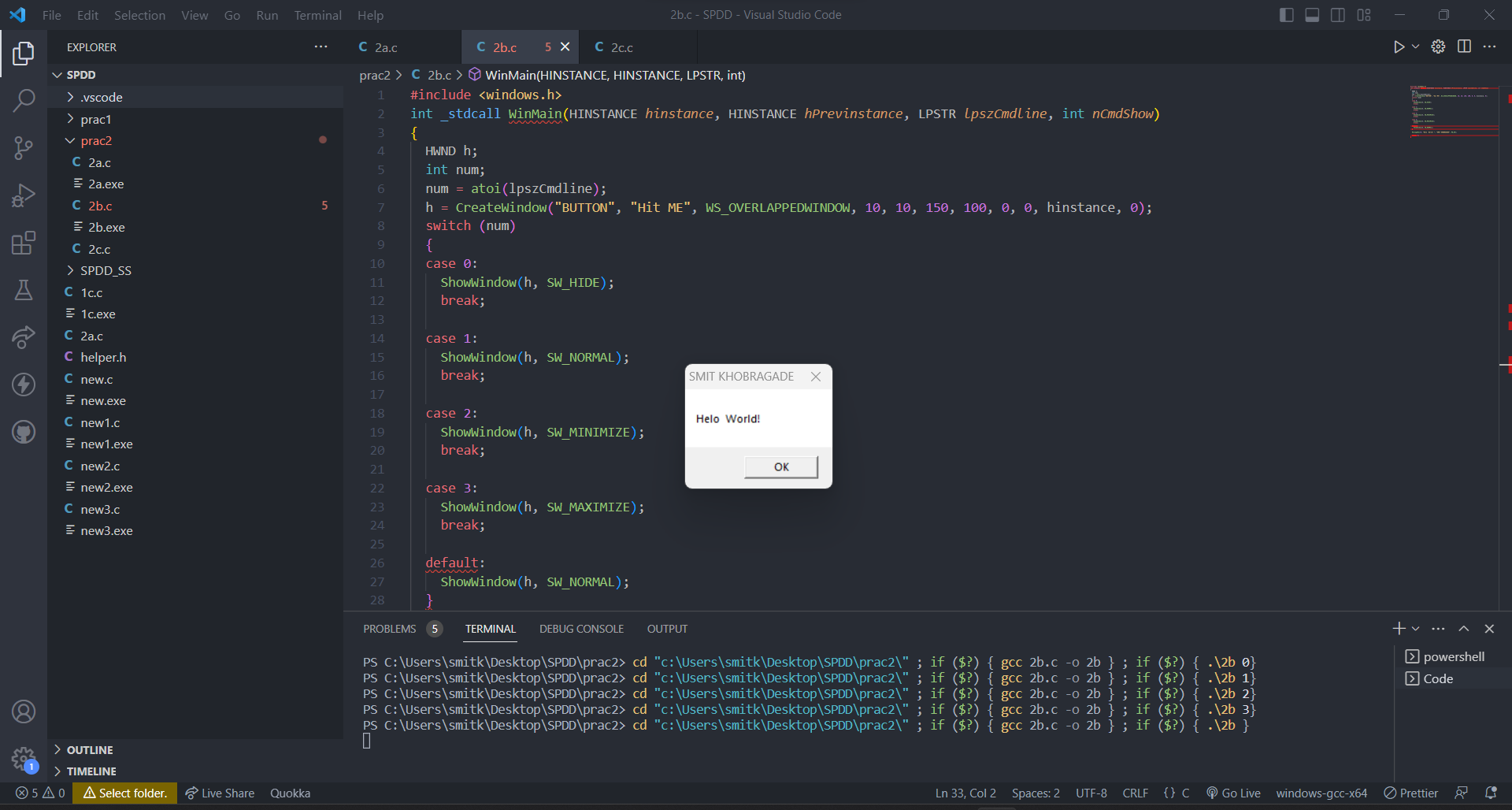
}

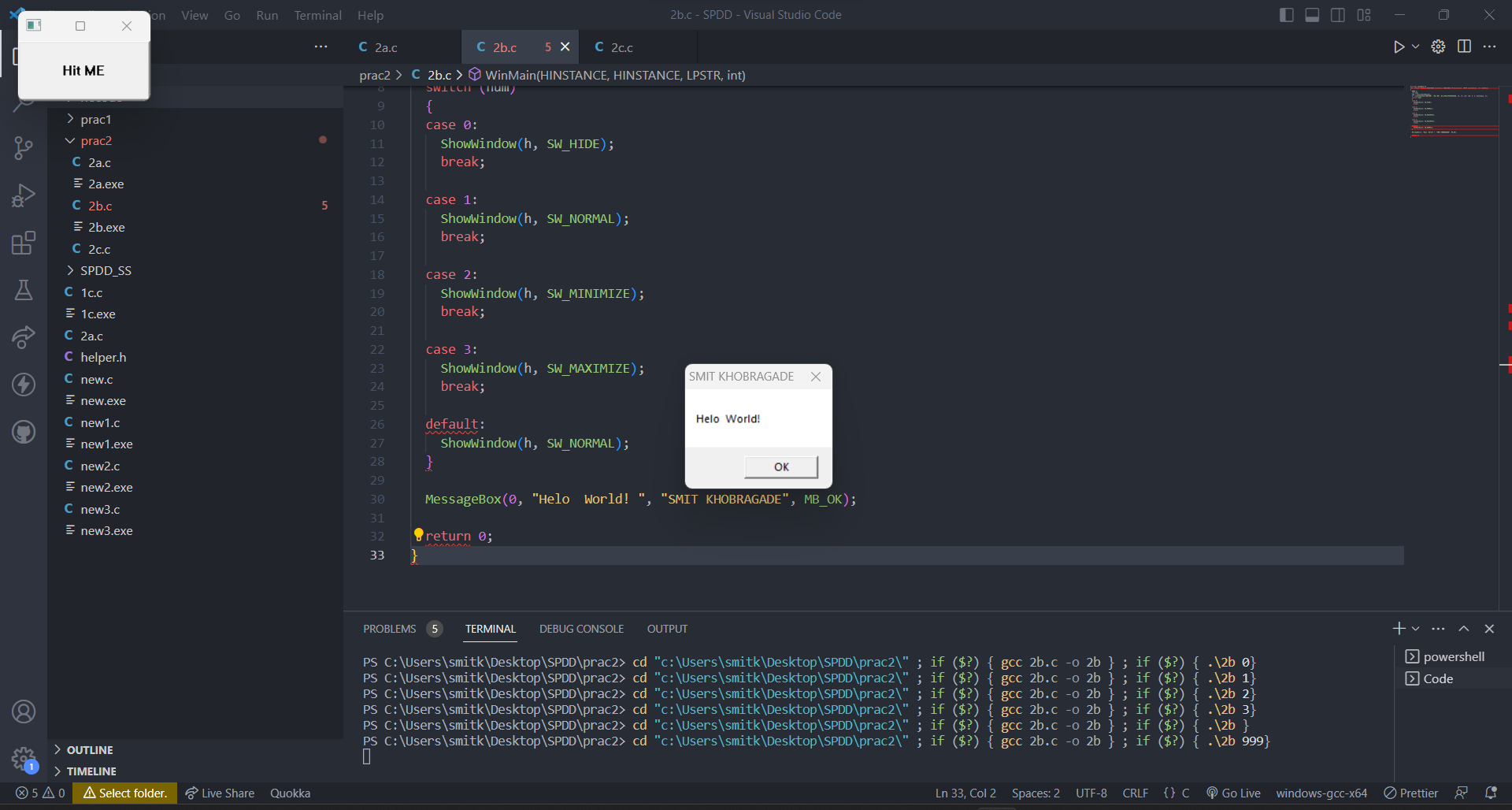












1. Write a program to create a simple window. Add the Menu bar, Menu item & Submenus to the created Window application and write a program to add some functionalities to the created menu items like save or open, etc.

#include <windows.h>

WNDCLASS wc;

#define ID\_FILE\_NEW 1

#define ID\_FILE\_OPEN 2

#define ID\_FILE\_EXIT 3

void addMenu(HWND *h*);

long \_stdcall myfunc(HWND *w*, UINT *x*, UINT *y*, long *z*)

{

  switch (x)

  {

  case WM\_DESTROY:

    PostQuitMessage(0);

    printf("Executed Successfully");

    break;

  case WM\_CREATE:

    addMenu(w);

    break;

  case WM\_COMMAND:

    switch (y)

    {

    case ID\_FILE\_NEW:

      MessageBox(0, "New button clicked", "File", MB\_OK);

      break;

    case ID\_FILE\_OPEN:

      MessageBox(0, "Open button clicked", "File", MB\_OK);

      break;

    case ID\_FILE\_EXIT:

      MessageBox(0, "Exit button clicked", "File", MB\_OK);

      break;

    }

  default:

    return DefWindowProc(w, x, y, z);

  }

  return 0;

}

void addMenu(HWND *h*)

{

  HMENU hMenubar;

  HMENU hMenu, hMenu1;

  hMenubar = CreateMenu();

  hMenu = CreateMenu();

  // hMenubar1 = CreateMenu();

  hMenu1 = CreateMenu();

  HMENU hSubMenu = CreatePopupMenu();

  AppendMenuW(hMenu, MF\_STRING | MF\_POPUP, (UINT\_PTR)hSubMenu, L"&New");

  AppendMenuW(hSubMenu, MF\_STRING, ID\_FILE\_NEW, L"Empty &File");

  AppendMenuW(hSubMenu, MF\_SEPARATOR, 0, NULL);

  AppendMenuW(hSubMenu, MF\_STRING, ID\_FILE\_NEW, L"Class..");

  AppendMenuW(hMenu, MF\_STRING, ID\_FILE\_OPEN, L"&Open");

  AppendMenuW(hMenu, MF\_SEPARATOR, 0, NULL);

  AppendMenuW(hMenu, MF\_STRING, ID\_FILE\_EXIT, L"&Quit");

  AppendMenuW(hMenubar, MF\_POPUP, (UINT\_PTR)hMenu, L"&File");

  AppendMenuW(hMenu1, MF\_STRING, ID\_FILE\_EXIT, L"&Undo");

  AppendMenuW(hMenubar, MF\_POPUP, (UINT\_PTR)hMenu1, L"&Edit");

  SetMenu(*h*, *hMenubar*);

}

int \_\_stdcall WinMain(HINSTANCE *hInstance*, HINSTANCE *hPrevInstance*, LPSTR *lpszCmdline*, int *nCmdShow*)

{

  HINSTANCE hinst;

  HWND hwndMain;

  MSG msg;

  wc.lpfnWndProc = myfunc;

  wc.hInstance = hInstance;

  wc.lpszClassName = "Main Class";

  wc.lpszMenuName = "First Menue";

  RegisterClass(&wc);

  hwndMain = CreateWindow("Main Class", "SMIT KHOBRAGADE", WS\_OVERLAPPEDWINDOW, 0, 0, 300, 300, 0, 0, hInstance, 0);

  ShowWindow(hwndMain, SW\_NORMAL);

  while (GetMessage(&msg, 0, 0, 0))

  {

    DispatchMessage(&msg);

  }

  return 0;

}

