```cpp
/*==============================================================================
 PROGRAMMER:            Brett Story,    Kyle Falconer
 FOLDERS:               Brett322,       Falconer1
 BRETT'S TASKS:         Patterned polygon, pixelmap, storing file (Contribution: 50%)
 KYLE'S TASKS:          Circle, Bitmap, text    (Contribution: 50%)
 COURSE:                CSC 525/625
 MODIFIED BY:           N/A
 LAST MODIFIED DATE:    Oct. 5, 2013
 DESCRIPTION:           Demo: drawing points.
 NOTE:                  Alpha transparencies were implemented below the desk to give
                        a shadow look. The image in the background was loaded from a
                        C struct, which the image editing program GIMP can export to.

                        We worked on the assignment together in person and completed
                        an equal number of tasks, alloted to about 50% each.

 FILES:                 h3.cpp, (hwProject.sln, ...)
 IDE/COMPILER:          MicroSoft Visual Studio 2012
 INSTRUCTION FOR COMPILATION AND EXECUTION:
    1.      Double click on labProject.sln  to OPEN the project
    2.      Press Ctrl+F7                   to COMPILE
    3.      Press Ctrl+Shift+B              to BUILD (COMPILE+LINK)
    4.      Press Ctrl+F5                   to EXECUTE
==============================================================================*/
#define _USE_MATH_DEFINES
#include <iostream>
#include <GL/glut.h>                // include GLUT library
#include <cmath>                    // include math library
#include <string>
using namespace std;

// To allow file reading
#include <fstream>
using std::ifstream;

// To exit if the file doesn't exist
#include <cstdlib>

//******************************************************************************

GLfloat PixelsRead[786432];

// Is the pattern used for the wall paper
GLubyte WallPaperPattern[128] = {0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00,

    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,

    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,

    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,

    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,

    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,
    0X00, 0XFF, 0XFF, 0X00,

    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,
    0X00, 0X0F, 0XF0, 0X00,

    0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00,
    0X00, 0X00, 0X00, 0X00};

// Is a c struct of the image behind the computer
static const struct {
  unsigned int   width;
  unsigned int   height;
  unsigned int   bytes_per_pixel; /* 2:RGB16, 3:RGB, 4:RGBA */
  unsigned char  pixel_data[32 * 48 * 3 + 1];
} gimp_image = {
  32, 48, 3,
  "]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f"
  "]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f"
  "]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f"
  "]\327f]\327f]\327f]\327f]\327f\327\312]\327\312]\327\312]]\327f]\327f]\327"
  "f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327"
  "f]\327f]\327f]\327f]\327f]\327f]\327f]\327f]\327f\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]]\327f]\327"
  "f]\327f]\327f]\327f\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]]\327f]\327f]\327f]\327f]\327"
  "f\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
  "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
```

```
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\315_\327\323"
    "a\327\315]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\315_\327\323_\327\315]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\315_\273\260S?<\35\313"
    "\277Y\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\304\270VLI(\302\267V\327\315_\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\315_\247\235I\35\34\17\5\5\5:8\34\327\320_\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\320_>;$\25"
    "\25\25.,\36\260\246P\327\315_\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\276\263T\26\25\14\1\1\3\5\5\5\7\7\7\216\207>\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\207\201?\24\24\24\25\25\25\24\24\26"
    "+)\37\304\273V\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\315],*\26\2\2\4\5"
    "\5\5\6\6\6\7\7\7\33\33\21\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\323\306[#\"\31\25\25\25\27\27\27\31\31\31\30\30\32FC*\327\315]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\250\236J\2\2\4\5\5\5\6\6\6\10\10\10\11\11\11\12\12"
    "\12JF$\327\315_\327\312]\327\312]\327\312]\327\315_HE(\25\25\25\30\30\30"
    "\31\31\31\33\33\33\34\34\34\35\35\35\263\251Q\327\312]\327\312]\327\312]"
    "\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\315_"
    "IE!\5\5\5\7\7\7\10\10\10\11\11\11\12\12\12\14\14\14\20\20\16\240\226F\327"
    "\315_\327\315_\327\315_\227\220E\30\30\26\30\30\30\31\31\31\32\32\32\34\34"
    "\34\35\35\35\37\37\37fb8\327\315]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\315]\10\10\6\6\6\6\10\10\10"
    "\11\11\11\13\13\13\14\14\14\16\16\16\20\20\20\271\256Q\247\237K\206\200>"
    "\255\242M\261\251O\26\26\26\32\32\32\33\33\33\35\35\35\36\36\36\37\37\37"
    "!!!++%\327\315]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\317\303[\1\1\3\6\6\10\7\7\11\12\12\12\12\12\14\13"
    "\13\15\14\14\16\203~=zu:\40\37\30\36\35\30##\33\207\201Aa]3\30\31\32\33\33"
    "\35\34\34\36\37\37\37!!!\"\"\"\31\32!\323\307]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\316\302Z\17\17\13"
    "\33\33\21\34\34\22\36\35\24\37\36\25!!\27#\"\27\316\302Z\31\31\25\23\23\25"
    "\27\27\27\30\30\30\40\40\34\302\267V,,\"..$//%10'21(33)++'\323\307]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\326\311^\306\272V\307\273W\307\273W\307\273W\307\273W\307\273W\307\273"
    "W\324\310^\14\15\22\27\27\27\31\31\31\32\32\32\24\24\30\324\310^\310\274"
    "X\310\274X\310\274X\311\275Y\311\275Y\311\275Y\310\275Z\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]to8\21\22\25\26\26\30\24\24\30\177z?\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\324\307\\\247\237K\204\177@\253\243M\324\310^\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\216\210D><(SP1CA+\223\214E\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\324\310^\"!\36\37\37\37\""
    "\"\40\"\"\"('$\323\307]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]uq;\36\36\36\40\40\40!!!###%%%ea9\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\315\304"
    "[\36\36\36\40\40\40\"\"\"###$$$&&&'''\301\266W\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]]Y3\40\40\40\"\"\"###$$$&&&''')))A@"
    "1\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312"
    "]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\300\265V\35\35\37"
    "!!!###$$$&&&''')))******\234\226L\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\305\274WZW4\"!#\"#$$$&%&'((()))((*MK5\254\242P\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\324\310^\301\266W\216\210Fpk>eb;lg>\206"
    "\201F\266\254V\324\310^\326\311^\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\326\311^\325\310]\325\310]\325\310]\326"
    "\311^\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327"
    "\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]\327\312]",
  };

  #define win_width 32
  #define win_height 32
  GLubyte win_logo[128] = {
      0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0xfc, 0x00,
      0xf0, 0x3f, 0xf3, 0x00,
      0xff, 0xff, 0x03, 0xc0,
      0xff, 0xfc, 0x00, 0xc0,
      0xcf, 0xcc, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0xfc, 0xc0,
      0xf0, 0x3c, 0xff, 0xc0,
      0xff, 0xff, 0xff, 0xc0,
      0xff, 0xff, 0x03, 0xc0,
      0xff, 0xfc, 0x00, 0xc0,
      0xcf, 0xcc, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0x00, 0xc0,
      0xc0, 0x0c, 0xfc, 0xc0,
      0xf0, 0x3c, 0xff, 0xc0,
      0xff, 0xff, 0xff, 0xc0,
      0x1f, 0xff, 0x03, 0xc0

  };



  void drawPoints()
  {
      // Following section draws a polygon pattern as background wall paper
      glEnable(GL_POLYGON_STIPPLE);                //Enables polygon stipple
      glPolygonStipple(WallPaperPattern);          //Loads custom pattern
      glBegin(GL_POLYGON);
      glColor3f(0.5, .6, 0);                       //Creates polygon from vertices
      glVertex2i(-300, 300);                       //and changes color for each point.
      glColor3f(0.5, 0.5, .1);
```

```
        glVertex2i(-300, -300);
        glColor3f(0.5, .6, 0);
        glVertex2i(300, -300);
        glColor3f(0.5, 0.5, .1);
        glVertex2i(300, 300);
        glEnd();
        glDisable(GL_POLYGON_STIPPLE);

        // Following draws shadow beneath desk
        glPolygonMode(GL_CCW, GL_FILL);             //Changes mode to fill
        glEnable(GL_BLEND);                         //Enables alpha blending
        glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
        glBegin(GL_POLYGON);
        glColor4f(0, 0, 0, .75);
        glVertex2i(-300, -175);
        glVertex2i(300, -175);
        glVertex2i(300, -300);
        glVertex2i(-300, -300);
        glEnd();
        glDisable(GL_BLEND);                        //Disables Alpha blending

        // Following section draws a rectangle for the desk surface

        glBegin(GL_POLYGON);                            //Creates polygon from vertices
        glColor3f(.5, .3, .1);                      //and changes color for each point.
        glVertex2i(-300, -50);
        glVertex2i(300, -50);
        glVertex2i(300, -150);
        glVertex2i(-300, -150);
        glEnd();

        // Following draws dark area of desk front
        glBegin(GL_POLYGON);
        glColor3f(.4, .2, .05);
        glVertex2i(-300, -150);
        glVertex2i(300, -150);
        glVertex2i(300, -175);
        glVertex2i(-300, -175);
        glEnd();

        // Following draws line on desk for psuedo-perspective
        int CurrentLineStartX = 600;
        int TotalSurfaceLines = 12;
        glLineWidth(4);
        for (int i = 0; i < TotalSurfaceLines; i++){
            glBegin(GL_LINES);
            glColor3f(.45, .25, 0.08);
            glVertex2i(CurrentLineStartX, -50);
            glVertex2i(CurrentLineStartX - 400, -150);
            glEnd();
            CurrentLineStartX -= 100;
        }

        glBegin(GL_LINES);
```

```
    glVertex2i(-300, -50);
    glVertex2i(300, -50);
    glEnd();
    glBegin(GL_LINES);
    glVertex2i(-300, -150);
    glVertex2i(300, -150);
    glEnd();

    // Draws a multicolor bitmap
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);

    glRasterPos2i(100, 0);
    glPixelZoom(5, 5);
    glDrawPixels(32, 48, GL_RGB, GL_UNSIGNED_BYTE, gimp_image.pixel_data);
    glPixelZoom(1, 1);

    //Following draws computer monitor w/o screen
    glBegin(GL_POLYGON);
    glColor3f(0.6, 0.6, 0.5);
    glVertex2i(150, -100);
    glVertex2i(150, 100);
    glVertex2i(200, 100);
    glVertex2i(200, -75);
    glEnd();

    glBegin(GL_POLYGON);
    glColor3f(0.8, 0.8, 0.7);
    glVertex2i(-50, -100);
    glVertex2i(150, -100);
    glVertex2i(150, 100);
    glVertex2i(-50, 100);
    glEnd();

    //Following draws buttons onto the computer monitor bezel
    float radius = 1;
    float x_offset = 130;
    float y_offset = -62;
    glBegin(GL_POINTS);
    glColor3f(0.85, 0.55, 0.55);
    for (int x=0; x<360; x++){
        float radians = x*(M_PI/180);
        glVertex2f(cos(radians)*radius+x_offset, sin(radians)*radius+y_offset);
    }
    glEnd();


    radius = 4.5;
    x_offset = -31;
    y_offset = -62;
    glBegin(GL_POINTS);
    glColor3f(0.6, 0.6, 0.6);
    for (int x=0; x<360; x++){
        float radians = x*(M_PI/180);
        glVertex2f(cos(radians)*radius+x_offset, sin(radians)*radius+y_offset);
    }
    glEnd();

    radius = 4;
    x_offset = -32;
    y_offset = -62;
    glBegin(GL_POINTS);
    glColor3f(0.65, 0.65, 0.65);
    for (int x=0; x<360; x++){
        float radians = x*(M_PI/180);
        glVertex2f(cos(radians)*radius+x_offset, sin(radians)*radius+y_offset);
    }
    glEnd();



    //Following draws screen onto computer monitor (black)
    glBegin(GL_POLYGON);
    glColor3f(0, 0, 0);
    glVertex2i(-25, -50);
    glVertex2i(125, -50);
    glVertex2i(125, 75);
    glVertex2i(-25, 75);
    glEnd();

    //Adds detail lines to computer monitor
    glBegin(GL_LINES);
    glColor3f(0.6, 0.6, 0.5);
    glVertex2i(-50, -50);
    glVertex2i(150, -50);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(125, 75);
    glVertex2i(149, 99);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(-25, 75);
    glVertex2i(-49, 99);
    glEnd();

    // Add text to the computer monitor
    glColor3f(0.0, 0.75, 0.0);
    glRasterPos2i(-10, 57);
    std::string s = "blargh";
    void * font = GLUT_BITMAP_8_BY_13;
    for (std::string::iterator i = s.begin(); i != s.end(); ++i)
    {
        char c = *i;
        glutBitmapCharacter(font, c);
    }


```

```
    // Draw an artistic version of the windows logo onto the computer monitor
    glPixelStorei(GL_UNPACK_ALIGNMENT, 4);
    glRasterPos2i(0, 0);
    glBitmap( win_width, win_height,   -20, 20,   0, 0,  win_logo);


    //This section begins the file saving
    glPixelStoref(GL_UNPACK_ALIGNMENT, 8);
    glReadPixels(0, 0, 512, 512, GL_RGB, GL_FLOAT, PixelsRead);

    // Removes any previous version of savedImg.txt
    remove("C:\\TEMP\\savedImg.txt");

    ofstream ResultFile("C:\\TEMP\\savedImg.txt");

    for (int i = 0; i < 786432; i++)
    {
        ResultFile << PixelsRead[i] << " ";
    }

    // Closes the file so it is no longer streaming
    ResultFile.close();

    cout << "File has been saved in C:\\TEMP\\savedImg.txt";
}


//************************************************************************************
void myInit()
{
    glClearColor(0, .3, .4, 0);         // specify a background clor: blueish-green
    gluOrtho2D(-300, 300, -300, 300);  // specify a viewing area
}

//************************************************************************************
void myDisplayCallback()
{
    glClear(GL_COLOR_BUFFER_BIT);   // draw the background

    drawPoints();

    glFlush(); // flush out the buffer contents
}

//************************************************************************************
void main(int argc, char ** argv)
{
    glutInit(& argc, argv);

    glutInitWindowSize(512, 512);               // specify a window size
    glutInitWindowPosition(100, 0);         // specify a window position
    glutCreateWindow("Simple Point Drawing");   // create a titled window

    myInit();                                   // setting up

    glutDisplayFunc(myDisplayCallback);     // register a callback

    glutMainLoop();                         // get into an infinite loop
}

```