

# Bài tập 2

Học kỳ 2, năm học 2017-2018

**Lưu ý:** Tất cả các bài tập đều phải VIẾT TAY

**Câu 1: Cho hàm `drawArcs()` như sau**

```
#define PI 3.1415926
void drawArcs(){
    glColor3f(0, 0, 0);
    glLineWidth(1);
    float R = 0.5;
    glBegin(GL_LINE_STRIP);
        for (int i = 0; i <= 90; i++){
            float x = R*cos(i*PI / 180.0);
            float y = R*sin(i*PI / 180.0);
            glVertex2f(x, y);
        }
    glEnd();
    glBegin(GL_LINE_STRIP);
        for (i = 180; i <= 270; i++){
            float x = R*cos(i*PI / 180.0) + 1;
            float y = R*sin(i*PI / 180.0) + 1;
            glVertex2f(x, y);
        }
    glEnd();
}
```

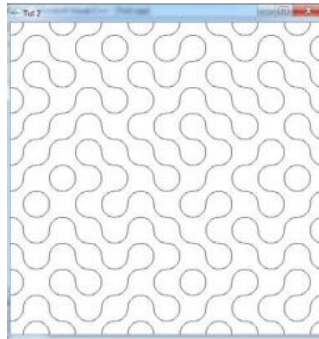
Hãy cho biết chức năng của hàm trên.

**Câu 2: Lát nền cửa sổ ứng dụng**

Người ta thường lát nền bằng cách, chọn một hình ảnh mẫu, sau đó sử dụng một tập các khung nhìn, cái nọ cạnh cái kia, che phủ bề mặt hiển thị, và hình ảnh mẫu được vẽ trong những khung nhìn đó. Một biến thể của cách làm trên là lựa chọn các hình mẫu một cách ngẫu nhiên từ tập các hình mẫu cho trước. Hãy sử dụng hàm `drawArcs()` ở trên, cùng các hàm `setViewport()` và `setCamera()`, để tạo thành hình vẽ lát nền tương tự như hình 1 (thường là không giống hệt), biết rằng cửa sổ màn hình có kích thước 600×600.

```
void setCameraVolume(float l, float r, float b, float t){
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D((GLdouble)l, (GLdouble)r, (GLdouble)b, (GLdouble)t);
}
```

```
void setViewport(int l, int r, int b, int t){
    glViewport(l, b, r - l, t - b);
}
```



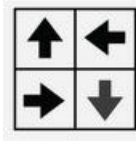
Hình 1

### Câu 3: Lát nền cửa sổ ứng dụng

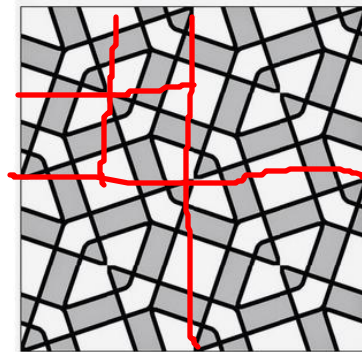
Biết rằng hình ảnh mẫu giống như hình 2.a, cách bố trí các hình ảnh mẫu giống như hình 2.b. Hãy viết chương trình để vẽ được hình vẽ tương tự hình 2.c. Chỉ cần viết vào giấy làm bài các hàm dùng để vẽ hình vẽ.



Hình 2.a



Hình 2.b



Hình 2.c

### Câu 4: Áp dụng các phép toán vector để tìm tâm hình cầu ngoại tiếp tứ diện

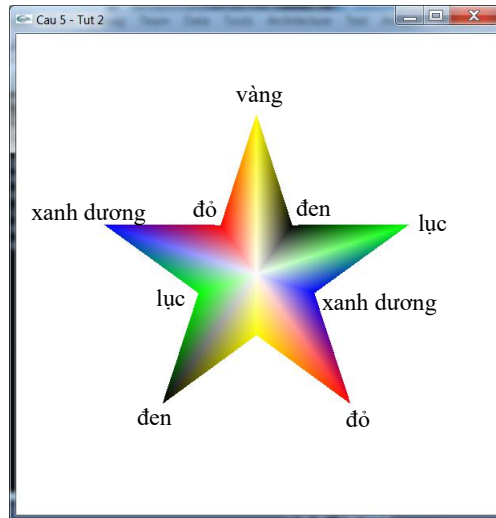
Cho tứ diện SABC với tọa độ các đỉnh như sau:

$$S = (10, 30, 10); A = (20, 0, 20); B = (0, 0, -30); C = (-30, 0, 0).$$

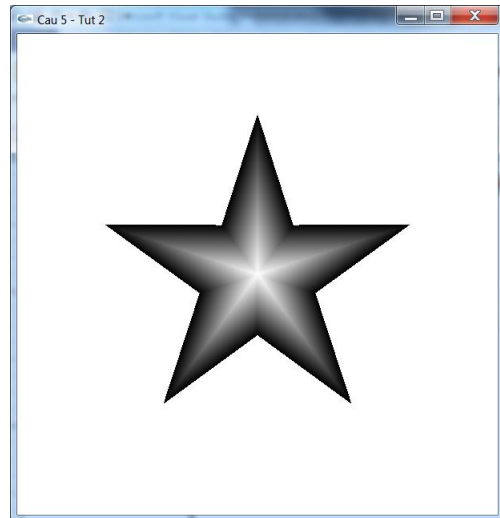
- Trong mặt phẳng  $y = 0$ , hãy viết phương trình tham số đường trung trực của hai cạnh AB và AC
- Trong mặt phẳng  $y = 0$ , hãy tìm tâm hình tròn ngoại tiếp tam giác ABC bằng cách tìm giao điểm của hai đường trung trực trên
- Viết phương trình đường thẳng của trục đường tròn ngoại tiếp tam giác ABC (tức là đường thẳng đi qua tâm đường tròn ngoại tiếp tam giác và vuông góc với mặt phẳng chứa tam giác)
- Xác định tâm của hình cầu ngoại tiếp tứ diện SABC.

### CÂU 5 : Sinh viên tham khảo chương trình demo đính kèm

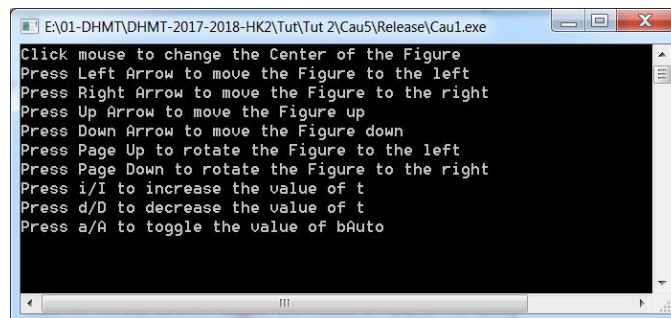
Viết chương trình vẽ hình ngôi sao, màu sắc các đỉnh của nó phụ thuộc vào tham số  $t$  ( $t$  thuộc  $[0, 1]$ ). Khi  $t = 1$ , tức là thời điểm ban đầu, màu sắc ban đầu của các đỉnh được cho trong hình vẽ 5.a. Khi  $t = 0$ , màu sắc tất cả các đỉnh đều là màu đen, xem hình 5.b.



Hình 5.a



Hình 5.b



Hình 5.c

Đoạn mã khởi động được cho trong phần phụ lục. Chương trình có khai báo các biến toàn cục như sau:

```
int      nWidth  = 600;
int      nHeight = 600;

int      nCenterX = nWidth/2;
int      nCenterY = nHeight/2;

int      nRadius  = 200;

float    vertexArr[10][2];

float    t = 1;

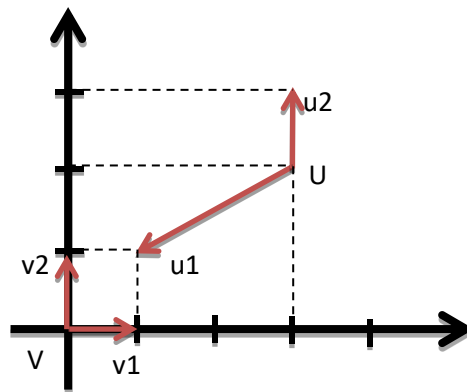
bool     bAuto = false;
```

Trong đó,  $(nCenterX, nCenterY)$  là vị trí của tâm hình vẽ,  $nRadius$  là bán kính hình tròn đi qua 5 đỉnh của hình vẽ,  $vertexArr$  chứa vị trí các đỉnh của hình ngôi sao. Khi  $bAuto$  bằng **true**, hình vẽ sẽ tự động thay đổi màu sắc các đỉnh từ màu sắc ban đầu (tham khảo hình 5.a) dần thành màu đen, và ngược lại. Khi  $bAuto$  bằng **false**, người sử dụng có thể sử dụng phím bấm để thay đổi giá trị của  $t$ . Chức năng của chương trình như hình vẽ 5.c.

Yêu cầu: Sinh viên hoàn thiện các hàm  $CalcVertexArrs()$ ,  $DrawFigure()$ ,  $OnSpecialKey()$ ,  $OnTimer()$ ,  $OnKey()$ ,  $OnMouseClicked()$  và viết các hàm này vào giấy làm bài.

### Câu 6: Biến đổi hệ trục tọa độ

Cho hệ trục tọa độ hai chiều thứ nhất ( $V, v_1, v_2$ ) với các vector cơ sở là  $v_1 = [1, 0]$ ,  $v_2 = [0, 1]$ . Cho một hệ trục tọa độ hai chiều thứ hai ( $U, u_1, u_2$ ) với các vector cơ sở  $u_1, u_2$  như trong hình vẽ, gốc tọa độ  $U$  ở vị trí  $(3, 2)$ .



Hình 6

- Một điểm A có tọa độ  $(2, 1)$  trong hệ trục tọa độ thứ hai. Hãy cho biết tọa độ của điểm A, trong hệ trục tọa độ thứ nhất.
- Một điểm B có tọa độ  $(3, 2)$  trong hệ trục tọa độ thứ nhất. Hãy cho biết tọa độ của điểm B trong hệ trục tọa độ thứ hai.
- Vector  $a$  được biểu diễn bằng  $(2, 3)$  trong hệ trục tọa độ thứ nhất. Hãy cho biết biểu diễn của vector này trong hệ trục tọa độ thứ hai.
- Vector  $b$  được biểu diễn bằng  $(2, 5)$  trong hệ trục tọa độ thứ hai. Hãy cho biết biểu diễn của vector này trong hệ trục tọa độ thứ nhất.

# PHỤ LỤC

Mã nguồn khởi động của câu 5.

```
#include "stdafx.h"
#include <iostream>
#include <Windows.h>
#include <glut.h>
#include <math.h>

using namespace std;

#define PI 3.1415926
#define DEGTORAD (PI/180.0)

int nWidth = 600;
int nHeight = 600;

int nCenterX = nWidth/2;
int nCenterY = nHeight/2;

int nRadius = 200;

float vertexArr[10][2];

float t = 1;

bool bAuto = false;

void setWindow(float l, float r, float b, float t)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(l, r, b, t);
}

void setViewport(int l, int r, int b, int t)
{
    glViewport(l, b, r-l, t -b);
}

void CalcVertexArrs()
{
}

void DrawFigure(float t)
{
}

void OnSpecialKey(int key, int x, int y)
{
}
```

```

void OnTimer(int value)
{

}

void OnKey(unsigned char key, int x, int y)
{

}

void OnMouseClicked(int button, int state, int x, int y)
{

}

void OnReshape(int w, int h)
{
    nWidth = w;
    nHeight = h;
}

void OnDisplay()
{
    CalcVertexArrs();

    glClearColor(1, 1, 1, 1);
    glClear(GL_COLOR_BUFFER_BIT);

    setViewport(0, nWidth, 0, nHeight);
    setWindow(0, nWidth, 0, nHeight);

    DrawFigure(t);

    glFlush();
}

int main(int argc, char** argv)
{
    cout << "Click mouse to change the Center of the Figure" << endl;
    cout << "Press Left Arrow to move the Figure to the left" << endl;
    cout << "Press Right Arrow to move the Figure to the right" << endl;
    cout << "Press Up Arrow to move the Figure up" << endl;
    cout << "Press Down Arrow to move the Figure down" << endl;
    cout << "Press Page Up to rotate the Figure to the left" << endl;
    cout << "Press Page Down to rotate the Figure to the right" << endl;
    cout << "Press i/I to increase the value of t" << endl;
    cout << "Press d/D to decrease the value of t" << endl;
    cout << "Press a/A to toggle the value of bAuto" << endl;

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(nWidth, nHeight);
    glutInitWindowPosition(20, 20);
    glutCreateWindow("Cau 5 - Tut 2");

    glutSpecialFunc(OnSpecialKey);
    glutKeyboardFunc(OnKey);
    glutReshapeFunc(OnReshape);
}

```

```
    glutMouseFunc(OnMouseClicked);  
    glutTimerFunc(300, OnTimer, 0);  
    glutDisplayFunc(OnDisplay);  
  
    glutMainLoop();  
    return 0;  
}
```