Đại học Bách khoa Đà Nẵng

Câu 1:

a. Viết thủ tục tô màu Fill(x,y,bc,c) bằng thuật toán tô tràn, với x,y là tọa độ thuộc miền tô màu. bc là màu viền còn c là màu vùng cần tô.
b. Nêu các hạn chế của thuật toán trên và cách giải quyết.

Câu 2:

a. Trình bày các bước vẽ đường cong Cn..... (mấy cái đường cong là thi hết đó, mỗi đề mỗi phần)
b. Viết chương trình nhập 2 số n,D,L sau đó vẽ đường cong đó

Câu 3:

a. Nêu định nghĩa phép affine 2 chiều
b. Chứng minh phép affine 2 chiều đảo bảm tỷ lệ chia đoạn thẳng (mỗi đề chứng minh 1 cái)

Thời gian 60 phút.
Đề không cho phép sử dụng tài liệu
Giáo viên coi thi không giải thích gì thêm.

Bài giải

<u>Câu 1:</u>

a.)

```
void TFloodfill(int x,int y,int bc,int c){
   if (getpixel(x,y)!=bc){
         putpixel(x,y,c);
      TFloodfill(x-1,y,bc,c);
      TFloodfill(x+1,y,bc,c);
      TFloodfill(x,y-1,bc,c);
      TFloodfill(x,y+1,bc,c);
   }
}
```

b)

<u>Hạn chế:</u>

+ Gọi đệ quy nhiều lần gây tràn stack
+ 1 điểm bị gọi lặp nhiều lần bởi các điểm kề nó->tô bị chậm

<u>Cách giải quyết:</u>

+ Hạn chế số lần gọi đệ quy (tô 3 điểm kề)
+ Không gọi đệ quy mà tô theo từng dòng

<u>Câu 2:</u>

a)

Các bước vẽ đường cong: (trúng đề nào thì viết phần đó

+ Koch: k_n độ dài l hướng d
  - Vẽ K_{n-1} độ dài l/3
  - Quay trái 60^o
  - Vẽ K_{n-1} độ dài l/3
  - Quay phải 120^o
  - Vẽ K_{n-1} độ dài l/3
  - Quay trái 60^o
  - Vẽ K_{n-1} độ dài l/3
+ C: C_n độ dài l hướng d
  - Quay trái 45^o
  - Vẽ C_{n-1} độ dài l*√2/2
  - Quay phải 90^o
  - Vẽ C_{n-1} độ dài l*√2/2
  - Quay trái 45^o để trả hướng
+ Rồng: C_n độ dài l hướng d và dấu s (-1 hoặc 1)
  - Quay trái s*45^o
  - Vẽ C_{n-1} độ dài l*√2/2
  - Quay phải s*90^o
  - Vẽ C_{n-1} độ dài l*√2/2

b) Chương trình: Trúng đề nào thì chép phần đó

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
#define Rad 0.017452
#define vuong 0.7071
// Duong cong Koch
void K(int n,float l,float d){
	if(n>0){
		K(n-1,l/3,d);d+=60;
		K(n-1,l/3,d);d-=120;
		K(n-1,l/3,d);d+=60;
		K(n-1,l/3,d);
	}
	else
	linerel(int(l*cos(d*Rad)),int(l*sin(d*Rad)));
}
// Duong cong C
void C(int n,float l,float d){
	if(n>0){
		d+=45;
		C(n-1,l*vuong,d);
		d-=90;
		C(n-1,l*vuong,d);
```

```
                d+=45;
        }
        else
                linerel(int(l*cos(d*Rad)),int(l*sin(d*Rad)));
}
// Duong cong Dragon
void CDragon(int n,float l,float d,int s){
        if(n>0){
                d+=45*s;
                CDragon(n-1,l*vuong,d,-1);
                d-=90*s;
                CDragon(n-1,l*vuong,d,1);
                d+=45;
        }
        else
                linerel(int(l*cos(d*Rad)),int(l*sin(d*Rad)));
}
// duong cong L
void L(int n,float l,float d){
        if(n>0){
                L(n-1,l/3,d);d+=90;
                L(n-1,l/3,d);d-=90;
                L(n-1,l/3,d);d-=90;
                L(n-1,l/3,d);d+=90;
                L(n-1,l/3,d);
        }
        else linerel(int(l*cos(d*Rad)),int(l*sin(d*Rad)));

}
void main(){
        int gd=0,gm=0;
        initgraph(&gd,&gm,"F:\\learn\\TC\\BGI");
         setcolor(LIGHTRED);
         // Koch
         outtextxy(10,0,"Cong  Koch: ");
         moveto(100,0);
         K(4,200,0);
         setcolor(LIGHTBLUE);
         // C
         outtextxy(10,100,"Cong  C: ");
         moveto(200,100);
         C(10,100,0);
         setcolor(LIGHTGREEN);
         // Dragon
         outtextxy(10,200,"Cong  Dragon: ");
         moveto(400,200);
```

```
    CDragon(10,100,0,1);
    setcolor(LIGHTGRAY);
    // Cong l
    outtextxy(10,300,"Cong  L ");
    moveto(400,300);
    L(4,200,0);
   getch();
   closegraph();
}
```

<u>Câu 3:</u>

a.

Đ/n:

Một phép biến đổi hai chiều sẽ biến đổi điểm P trong mặt phẳng thành điểm có tọa độ mới Q theo một quy luật nào đó. Về mặt bản chất, một phép biến đổi điểm là một ánh xạ T được định nghĩa :

$T: R^2 \to R^2$

$P(P_x,P_y) \to Q(Q_x,Q_y)$.

Có phương trình

$$\begin{cases} Q_x = aP_x + cP_y + Tr_x \\ Q_y = bP_x + dP_y + Tr_y \end{cases} \qquad ad-bc \# 0$$

b)

<u>Tỉ lệ chia đoạn thẳng:</u>

Phương trình tham số của đường thẳng đi qua 2 điểm A và B, với điểm C chia AB theo tỉ lệ t là:

$P = (1-t)A + tB$

Xét phép $T = (M,Tr)$: $P \to Q$, ta có:

T(C):

$Q = PM + Tr$

$=C*M + Tr$

$= [(1-t)A + tB]M + Tr$

$= (1-t)AM + tBM + Tr$

$= (1-t)(AM +Tr) + t(BM + Tr)$

Nếu gọi A’, B’,C’ lần lượt là ảnh của A, B,C qua phép biến đổi T, ta sẽ có $C'=(1-t)A' + tB'$

Vậy, điểm C’ cũng chia A’,B’ theo tỉ lệ t hay phép affine bảo toàn tỉ lệ chia đoạn thẳng.

<u>Tính thẳng hang:</u>

Phương trình tham số của đường thẳng đi qua 2 điểm A và B

$P = (1-t)A + tB$

Xét phép T = (M,Tr): P -> Q, ta có:
Q(t)=P(t)*M = [(1-t)A+tB]*M= (1-t)AM+tBM

Nếu gọi A', B' lần lượt là ảnh của A, B qua phép biến đổi T, ta sẽ có

A'=AM, B`=BM.

Lúc này Q(t)=(1-t)A' + tB' .

Đây chính là dạng của phương trình tham số đoạn thẳng qua A', B'.

Từ kết quả trên, để biến đổi một đoạn thẳng đi qua hai điểm A và B, ta chỉ cần áp dụng phép biến đổi cho hai điểm A, B rồi vẽ lại đoạn thẳng qua hai điểm mới.

Tính song song:
Pt đường thẳng qua A có vector chỉ phương t β
L1: P=A+tβ
L2: P=B+tβ
T=(M,Tr): P->Q
T(L1): Q=P*M + Tr
Q=A*M + Tr +tβ*M
Q=T(A) + tβ*M .(1)(Vì A*M+Tr là ảnh của A qua phép biến đổi T)
Từ đó suy ra:
T(L2):
Q=T(B) + tβ*M.(2)

Từ (1)(2) suy ra T(L1)//T(L2)
Nên phép Affine bảo toàn tính song song