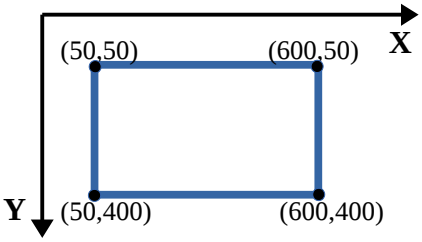
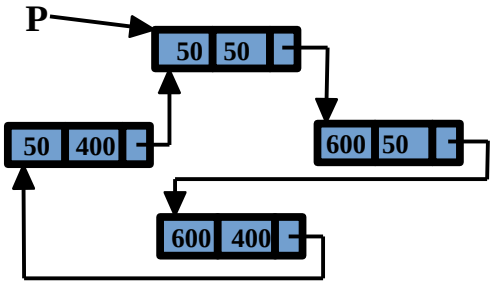
MAC122 Princípios de Desenvolvimento de Algoritmos
EP no. 3

Prof.: Paulo Miranda & Marco A. Gerosa
Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)

EP3 – Parte A:

Um polígono pode ser representado pela sequência de seus vértices, armazenados em uma lista circular em sentido horário. Exemplo:





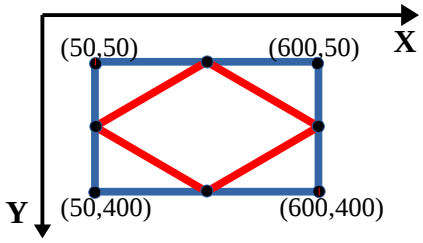
Considere a seguinte estrutura:

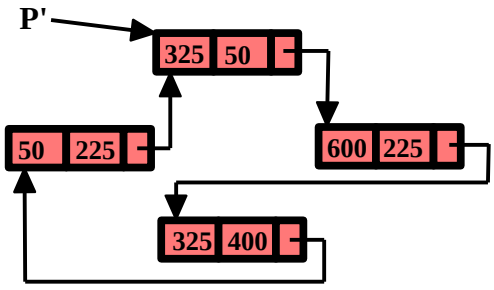
```
struct Vertice{
        float x;
        float y;
        struct Vertice *prox;
};
typedef struct Vertice* Poligono;
```

Você deve implementar as seguintes funções:

int NumeroDeVertices(Poligono P);
Função que conta o número de vértices de um polígono fornecido. Por exemplo, no caso da lista circular do exemplo acima, a função deve devolver 4.

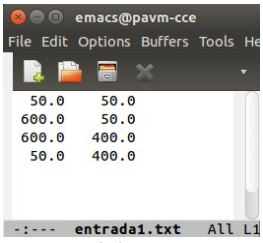
Poligono PoligonoPorPontoMedio(Poligono P);
Função que devolve um segundo polígono derivado de P, de tal modo que seus vértices são obtidos pelos pontos médios dos lados do primeiro polígono.
Exemplo:



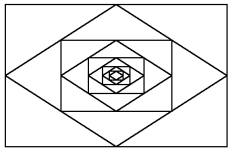


OBS: O polígono gerado deve ser armazenado em uma segunda lista circular que deve ser alocada dinamicamente. A função pode devolver o endereço de qualquer um dos vértices do polígono gerado (não existe uma ordem preferencial). A função deve devolver NULL se o polígono P for inválido, isto é, com menos que três vértices.

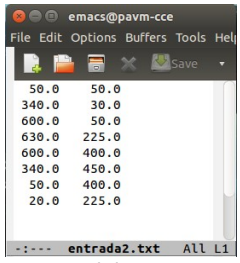
Faça um programa que lê um polígono de um arquivo texto fornecido, contendo as coordenadas de seus vértices (uma coordenada por linha), e que gera uma imagem de 640x480 contendo o desenho de uma sequência de polígonos por ponto médio, até um número máximo de repetições R. O nome do arquivo de saída deve ser "poligono.pgm".



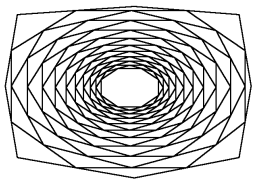
(a)



(b)



(c)



(d)

Figura 1: (a,c) Arquivos de entrada. (b,d) Imagem dos polígonos por ponto médio de a e c (com R=10 e 20 respectivamente).

EP3 – Parte B:

Um fractal é um objeto geométrico que pode ser dividido em partes, cada uma das quais semelhante ao objeto original. Fractais são muito usados em arte gerada por computador. O objetivo deste EP é fazer um programa em linguagem C, usando funções recursivas, para gerar imagens (no formato PGM) com os desenhos de alguns exemplos comuns de fractais, como a Estrela de Koch e a Curva de Lévy.

Estrela de Koch:

Este fractal, também conhecido como Floco de neve de Koch, inicia a sua construção a partir de um triângulo equilátero. Cada um dos seus segmentos de reta (lados do triângulo) é então submetido a alterações recorrentes, como a seguir se descreve:

1. Divide-se o segmento de reta em três segmentos de igual comprimento.
2. Desenha-se um novo triângulo equilátero, tendo como base o segmento central obtido no primeiro passo.
3. Apaga-se o segmento que serviu de base ao triângulo do segundo passo.

O processo é então repetido sucessivamente, para cada novo segmento obtido, até atingir um número máximo de iterações. O fractal é o limite das iterações ao infinito.

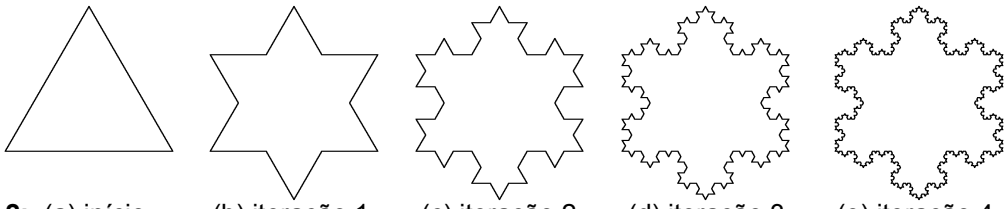


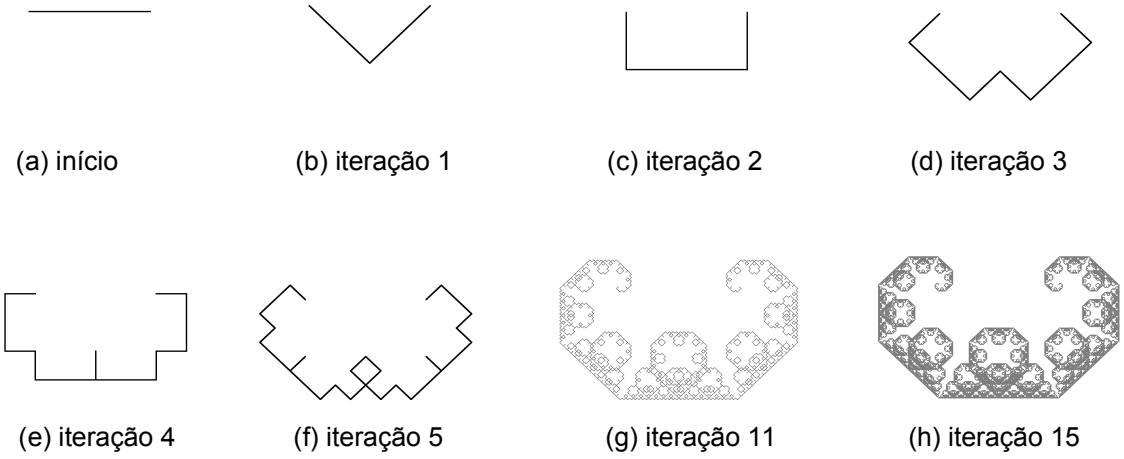
Figura 2: (a) início (b) iteração 1 (c) iteração 2 (d) iteração 3 (e) iteração 4

Curva de Lévy:

Em matemática a Curva de Lévy, também conhecida como Curva C, é um fractal autossimilar. A construção da curva C inicia-se com um segmento de reta (Figura 3a). Um triângulo isósceles com ângulos de 45°, 90° e 45° é construído usando esta linha como sua hipotenusa. A linha original é, então, substituída pelos dois outros lados deste triângulo (Figura 3b).

Na segunda etapa, cada uma das duas novas linhas forma a base de um outro triângulo isósceles em ângulo reto, sendo substituída pelos outros dois lados do seu respectivo triângulo. Assim, após duas etapas, a curva toma a aparência de três lados de um retângulo com o mesmo comprimento que a linha original, mas tendo apenas metade do valor como sua largura (Figura 3c).

Em cada etapa subsequente, cada segmento de reta da curva é substituído pelos outros dois lados de um triângulo isósceles retângulo construído sobre o segmento (Figuras 3d-h). Depois de n etapas a curva será composta por 2^n segmentos de reta, sendo cada um deles menor que a linha original por um fator de $2^{n/2}$.



(a) início (b) iteração 1 (c) iteração 2 (d) iteração 3

(e) iteração 4 (f) iteração 5 (g) iteração 11 (h) iteração 15

Figura 3: Iterações da Curva de Lévy.

Atividade:

Faça um programa em linguagem C que gera imagens (no formato PGM) dos desenhos dos dois tipos de fractais apresentados anteriormente, até um valor máximo de iterações fornecido pelo usuário. Na resolução do problema, o seu programa deve necessariamente usar funções recursivas.

Observações:
No caso do Estrela de Koch, utilize uma imagem quadrada (ex: 601x601). O triângulo equilátero inicial deve ser escolhido de modo que os seus vértices sejam equidistantes do centro da imagem (ex: a uma distância de 250 pixels do centro), e a base do triângulo deve ser paralela ao eixo x da imagem. O nome do arquivo de saída deve ser "estrela.pgm".

No caso da Curva de Lévy, utilize uma imagem quadrada (ex: 601x601). Considere a linha de referência inicial paralela ao eixo x da imagem e passando pelo seu centro. Considere também que os pontos extremos da linha de referência inicial são equidistantes do centro da imagem (ex: a uma distância de 128 pixels do centro). O nome do arquivo de saída deve ser "curvalevy.pgm".

Dica: utilize a função auxiliar abaixo que desenha um segmento de reta $\overline{AB}$ na imagem com o valor **`val`** fornecido. A imagem é dada pela matriz M com m linhas e n colunas, A = (x1,y1) e B = (x2,y2).

```
void DrawLine(int **M, int m, int n, int x1, int y1, int x2, int y2, int val){
  float x,y,dx,dy;
  int xi,yi;

  dx = (x2-x1);
  dy = (y2-y1);
  if(fabsf(dx) > fabsf(dy)){
    dy = dy/(fabsf(dx));
    dx = dx/(fabsf(dx));
  }
  else{
    dx = dx/(fabsf(dy));
    dy = dy/(fabsf(dy));
  }
  x = (float)x1;
  y = (float)y1;
  xi = x1;
  yi = y1;
  while(xi != x2 || yi != y2){
    if(yi >= 0 && yi < m && xi >= 0 && xi < n)
      M[yi][xi] = val;
    x += dx;
    y += dy;
    xi = (int)(x + 0.5);
    yi = (int)(y + 0.5);
  }
  if(y2 >= 0 && y2 < m && x2 >= 0 && x2 < n)
    M[y2][x2] = val;
}
```