Computação Gráfica
Primeira Avaliação - 2017.1

1- Calcule a matriz de uma transformação que corresponde ao espelhamento em trono da reta y=3x, seguida de uma rotação de 60º no sentido horário em torno do ponto (2,3), seguida de uma escala de (0.5, 3.2) em relação ao mesmo ponto.

2- Escreva um algoritmo que recebe as coordenadas de três pontos p1 (x1, y1), p2 (x2, y2)e p3(x3, y3) e retorna (-1, 0 ou 1)

-1 – caso o ponto p1 não esteja sobre o segmento de reta que liga p2 a p3 (caso de negativo trivial do algoritmo de pick de reta, só com a codificação dos vértices)

1 – caso o ponto p1 esteja esteja sobre o segmento de reta que liga p2 a p3 (caso de verdadeiro trivial do algoritmo de pick de reta, só com a codificação dos vértices)

0 – caso a codificação dos vértices não seja suficiente para determinar o pick de reta.

Obs: o algoritmo deve ser construído com a técnica de codificação dos vértices apresentada em aula pro algoritmo de pick de reta.

int VerificaPickReta(float x1, float x2, float x3, float y1, float y2, float y3)

1- Escreva um algoritmo que recebe as coordenadas de um conjunto de pontos ordenados angularmente em relação ao ponto p de coordenadas (x0, y0) (vetores de coordenadas x[1..n] e y[1..n]) (x[1] y[1] é o primeiro da ordenação, x[2] y[2] é o segundo da ordenação, e assim sucessivamente). E retorna o fecho convexo deste conjunto de pontos usando a ideia da marcha de Jarvis.

Obs: aqui não precisa colocar o algoritmo inteiro, algoritmo decorado inteiro sem o ajuste não vai servir.

```
typedef struct _ponto2D_ {
   float x[20000];  //vetor com coordenadas x dos pontos do fecho convexo
   float y[20000];  // vetor com coordenadas y dos pontos do fecho convexo
   int np;    // numero de pontos do gecho convexo
}Fecho
Fecho *JarvisDePontosOrdenados( int x[], int y[], int n)
```