USP – ICMC – SSC

SSC0501 - Introdução à Ciência da Computação I (Teórica)

Professor responsável: *Fernando Santos Osório* **E-mail:** fosorio @ icmc.usp.br
fosorio @ gmail.com
Semestre: 2015/1
Horário: Ter. 21h-22h40 e Sexta 19h-20h40 **Web:** ***http://www.icmc.usp.br/~fosorio/***

Nome do Aluno: _____

Número USP : _____ **DATA: 16 / 06 / 2015**

PROVA TEÓRICA FINAL – PF (Prova Escrita)

1) **Implemente o seguinte programa de visualização de primitivas gráficas, sendo que o programa tem 3 etapas a serem implementadas (cada etapa deve ser implementada por uma sub-rotina):**

1. **Ler um arquivo texto de configuração da aplicação gráfica "draw-cfg.txt"**
2. **Ler um arquivo texto com as primitivas gráficas de um arquivo ("desenho.txt") para vetor(es) em memória alocado dinamicamente**
3. **Exibir as primitivas gráficas na tela, usando as rotinas da biblioteca "graphics.h"**

Etapa 1. Ler um arquivo texto de configuração da aplicação gráfica "draw-cfg.txt" [3.0 pontos]
Faça uma sub-rotina para ler o arquivo de configuração que contém as seguintes informações: (1) Tamanho máximo (valor numérico inteiro) do vetor de primitivas gráficas (ver Etapa 2 sobre os dados das primitivas), que deve ser usado para alocar dinamicamente a estrutura de dados usada para guardas as primitivas em memória; (2) Sequência de seleção da tabela com 3 cores básicas usadas posteriormente para fazer os desenhos (cor 0, 1, 2), composta por índices seguidos dos valores RGB (inteiros), conforme exemplo abaixo. A sub-rotina que lê o arquivo de configuração deve ser uma função que retorna o tamanho máximo do vetor de primitivas lido do arquivo e deve preencher uma tabela (variável global externa) com os valores das cores.

Exemplo de Arquivo draw-cfg.txt (arquivo válido, com dados garantidamente sem erros)

```
1000
0  255 0 0
1  0 0 255
2  255 255 255
```

Este arquivo indica que teremos no máximo 1000 primitivas, e a tabela de cores é composta pelas cores: cor índice 0 com R=255 (red), G=0 (green) e B=0 (Blue); seguida pela cor índice 1 com R=0, G=0, B=255; e por fim a cor 2 com R=255,G=255, B=255. A variável externa é definida da seguinte forma (variável tipo "extern" declarada no arquivo graphics.h):

```
extern int TabCores[3][3];      //  TabCores[0][0] contém o R da cor índice 0
                                //  TabCores[0][1] contém o G e TabCores[0][2] contém o B da cor 0
```

Etapa 2. Ler um arquivo texto com as primitivas gráficas de um arquivo ("desenho.txt") para vetor(es) em memória alocado dinamicamente [3.0 pontos].
As primitivas gráficas usadas pelo programa visualizador de desenhos, são de quatro tipos, conforme descrição abaixo:
PONTO Xp Yp Cor => Exemplo: P 10 10 0 (ponto na coordenada (10,10) com cor índice 0)
LINHA Xi Yi Xf Yf Cor => Exemplo: L 20 20 40 40 1 (linha de (20,20) até (40,40) com cor 1)
RECT Xi Xi Xf Yf Cor => Exemplo: R 50 50 70 70 0 (retângulo de (50,50) até (70,70) cor 0)
CIRC Xc Yc Raio Cor => Exemplo: C 50 50 10 2 (círculo com centro em (50,50), raio 10 e cor 2)

Exemplo do Conteúdo do Arquivo "desenho.txt": (os dados do arquivo são corretos e perfeitos)

```
P 10 10 0
P 80 80 1
L 20 20 40 40 1
R 50 50 70 70 0
C 50 50 10 2
L 30 30 30 50 2
C 70 70 10 2
```

A **sub-rotina** que lê o arquivo de desenho deve ser uma função que **lê o arquivo texto em disco para um vetor em memória** e **retorna um valor inteiro** com o ***total de primitivas lidas*** do arquivo, no caso do exemplo acima retorna o valor 7 (7 primitivas de desenho), e também deve **retornar, em variáveis de parâmetros passadas por referência**, o valor do ***total de Pontos*** (2 no exemplo acima), ***de Linhas*** (2 no exemplo acima), ***de Retângulos*** (1 no exemplo acima) ***e de Círculos*** (2 no exemplo acima), retornando as variáveis nesta ORDEM (Pontos, Linhas, Retângulos e Círculos). No programa principal, este deverá exibir estes dados na tela e as depois permitir a visualização das primitivas (realizada na Etapa 3), conforme o exemplo abaixo.

Exemplo da tela de saída de execução do programa:

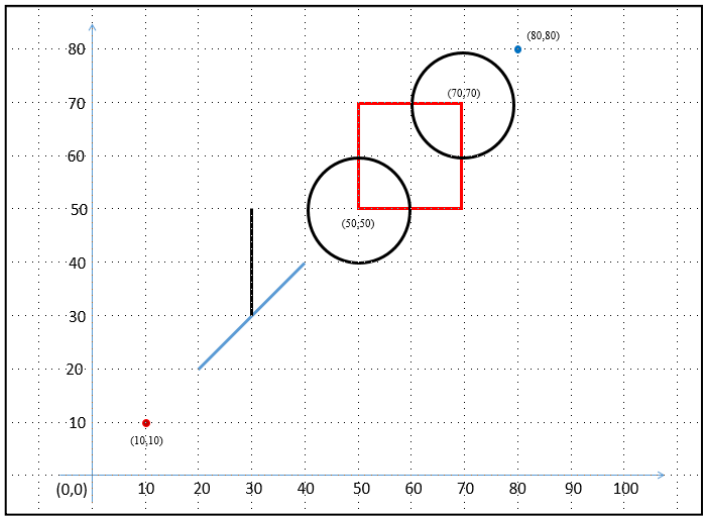
```
>> ACME Power Draw <<

Lendo Arquivo de Configuração...
Maximo de Primitivas: 1000

Lendo Arquivo de Primitivas...
Total de Primitivas: 7
Total de Pontos: 2
Total de Linhas: 2
Total de Retangulos: 1
Total de Circulos: 2

Desenhando...

>> Pressione uma tecla para sair do programa
```



Etapa 3. Exibir as primitivas gráficas na tela, usando as rotinas da biblioteca "graphics.h" [3.0 pontos]
Esta sub-rotina realiza a inicialização gráfica e o desenho das primitivas na tela gráfica usando as rotinas da biblioteca gráfica que estão disponíveis no "graphics.h". As rotinas gráficas a serem usadas são as seguintes: (graphics.h)

```
void initgraphics (void);                          // Inicializa o modo gráfico
void setcolor (int color);                         // Seleciona cor para Linha, Retangulo e Circulo
void putpixel (int x, int y, int color);           // Ponto
void line (int x1, int y1, int x2, int y2);        // Linha
void rectangle (int x1, int y1, int x2, int y2);   // Retangulo
void circle (int x, int y, int radius);            // Circulo
void writeimagedraw (const char* filename, int xmin, int ymin, int xmax, int ymax);
                                                   // Salva uma região da imagem em um arquivo
```

Para desenhar as primitivas na tela, primeiramente deve ser inicializado o modo gráfico (initgraphics) e depois devem ser exibidas as primitivas, uma a uma, chamando as respectivas sub-rotinas de desenho indicadas acima. Uma vez terminado o desenho na tela, salvar a imagem em disco usando a rotina "writeimagedraw", selecionando a região da imagem que será salva.

As imagens criadas pelo "graphics" possuem uma resolução máxima de 640x480 (640 de largura no eixo X e 480 de altura no eixo Y). Entretanto, deseja-se salvar apenas a área do desenho, ou seja, cabe a você determinar as coordenadas mínimas (Xmin e Ymin) e máximas (Xmax e Ymax) utilizadas pelas primitivas para realizar o desenho. De posse dos limites da área desenhada (Xmin,Ymin) e (Xmax,Ymax) você deve salvar em disco a imagem considerando apenas esta região da imagem, e usando a rotina "writeimagedraw" para salvar uma região da imagem.

ATENÇÃO:

- ⇨ A prova possui 3 etapas, mas se constitui de um **único programa COMPLETO** que deve incluir estas três etapas em seqüência.

- ⇨ Você deve ter percebido que a prova soma 3.0+3.0+3.0 pts por cada Etapa (9.0 pts. ao total). Um ponto da prova (1.0 pt) é reservado para uma avaliação geral da implementação: main, typedefs, structs, alocação dinâmica, sub-rotinas, parâmetros, variáveis locais e globais usadas, estrutura e lógica do programa; ou seja, a "qualidade geral do código".

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É **INDIVIDUAL.**
2. A PROVA É **COM CONSULTA AO MATERIAL INDIVIDUAL.**
   (Pode consultar: cadernos, anotações, livros – qualquer tipo de material escrito ou impresso)
3. **NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL** (Cadernos, Anotações, Livros, etc).
4. **NÃO É PERMITIDO O USO DE DISPOSITIVOS ELETRÔNICOS** durante a prova.
   (Não pode usar: notebook, computador, palmtops/pdas, celular, etc.)
5. RESPONDER A PROVA NAS FOLHAS FORNECIDAS: A CANETA OU A LÁPIS.
   SE FOR RESPONDIDA A LÁPIS E TIVER QUALQUER INDÍCIO DE ALTERAÇÃO OU RASURA, **PODEM NÃO SER ACEITOS PEDIDOS DE REVISÃO** DE PROVA.
6. LEMBRE-SE DE **IDENTIFICAR A PROVA** COM O SEU NOME E NÚMERO USP.
   DEVOLVER A FOLHA DE RESPOSTAS JUNTAMENTE COM A PROVA (Questões).
7. DURAÇÃO: Max. 3 horas
