Departamento de Ciência da Computação – IME-USP

MAC 115 – Introdução à Computação para Ciências Exatas e Tecnologia

ESCOLA POLITÉCNICA – PRIMEIRO SEMESTRE DE 1998

Segundo Exercício-Programa (complemento) Data de entrega: até 8 de maio de 1998.

Turmas: Todas.

GRÁFICOS

1 Introdução

No segundo semestre você deverá cursar a disciplina MAP 121 CÁLCULO NUMÉRICO. Em MAP 121 você deverá escrever programas que fazem gráficos de funções. Por esta razão estamos dando este exercício-programa complementar. Na realidade este exercício-programa será mais um trabalho de digitação e esperamos que ao você digitar e executar o programa abaixo você aprenda como fazer um programa que mostra na tela o gráfico de uma função. Este exercício complementar é facultativo e os alunos que o fizerem receberão 3 (três) pontos a mais na nota do segundo exercício-programa. (Quem tirar 10.0 no segundo exercício-programa e fizer este exercício corretamente ficará, para efeito da média do final do semestre, com nota 13.0 no segundo exercício-programa.)

Este exercício (excepcionalmente) deverá ser feito no `Quick C`. Entretanto outros compiladores `C` tem bibliotecas gráficas com funções semelhantes às que serão usadas.

2 O quê você deverá fazer

Neste exercício você fará um programa que mostra na tela o gráfico da função `seno`. Para isto digite o programa que está abaixo junto com a função `seno` que você escreveu no seu segundo exercício-programa (se você não escreveu a função `seno` para o seu segundo exercício-programa então você deverá escrevê-la para este). No programa abaixo a chamada da função `sin` da biblioteca padrão deverá ser substituída pela chamada da sua função `seno`.

A exemplo do que ocorreu no segundo exercício-programa, a soma do cáculo do `seno` deverá incluir os termos da série de potência até que o primeiro termo de valor absoluto $< 10^{-8}$ seja obtido—ou seja, o seno deverá ser calculado com precisão de 10^{-8}.

O programa abaixo está bem comentado. Assim não deverá ser muito difícil você entender o que faz a maior parte dos comandos gráficos (`_moveto(x,y)`: move o "pincel" para a posição `(x,y)` da tela sem traçar; `_lineto(x,y)`: move o "pincel" para da posição atual até a posição `(x,y)` da tela traçando um segmento de reta entre estas duas posições). No HELP do `Quick C` você encontra a descrição de cada uma das funções usadas pelo programa. Você **não** precisa se preocupar em entender tudo que o programa faz.

Apesar deste exercício parecer um simples trabalho de digitação você, se desejar, pode aprender mais sobre a linguagem `C` e também aprender sobre como trabalhar com o módulo gráfico do `Quick C` (aproveite!).

Para imprimir a saída gráfica do programa utilize a tecla $\boxed{\texttt{PRINT SCREEN}}$. É possível que durante ou após a impressão, ou melhor, depois que se pressiona a tecla $\boxed{\texttt{PRINT SCREEN}}$, o micro "trave". Assim, recomenda-se deixar a impressão do gráfico por último, quando tiver certeza de que o programa está funcionando com sua saída gráfica correta.

3 Listagem do Programa

Abaixo está a listagem do programa que você deverá digitar. O programa utiliza a biblioteca gráfica do `Quick C` disponível na rede do CCE. Estas funções gráficas não fazem parte da linguagem `C` padrão ANSI. Assim, se este programa funcionar com outra plataforma além do `Quick C`, isto será um verdadeiro milagre.

```
/*==============================================================*\
 *                 MAP121 - Cálculo Numérico                    *
 *--------------------------------------------------------------*
 *   Arquivo....: GRAPH.C                                       *
 *   Autor......: Sérgio Moriaki Oshiro                         *
 *   Versão.....: 1.0 [07/11/96]                                *
 *   Descrição..: Este arquivo é um exemplo de utilização das   *
 *                funções gráficas  do  Quick C  existente na   *
 *                rede do CCE. Ele é uma adaptação do progra-   *
 *                ma SAMPLE.C escrito para o Borland C.         *
 *--------------------------------------------------------------*
 *   Complilador: Quick C v2.5                                  *
\*==============================================================*/

/*---------------------------------------------------------------
   Para uma descrição mais detalhada das funções e  constantes
   utilizadas ao longo do programa, posicione o cursor  sob  o
   nome desejado e pressione a tecla F1.
 --------------------------------------------------------------*/

#include <graph.h>   /* "pacote" gráfico */
#include <stdlib.h>  /* "pacote" com tipos, variáveis e funções usuais  */
#include <conio.h>   /* "pacote" de entrada/saída via console do MS-DOS */
#include <stdio.h>   /* "pacote" de entrada/saída padrão */
#include <math.h>    /* "pacote" matemático */

#define FONPATH "E:\\QC25\\SAMPLES\\*.FON" /* caminho para as fontes */
#define MARGINS 50                   /* margens do gráfico em pixels */
#define TITLE   "Grafico: Seno"                /* título do gráfico */

/* protótipo da função "plota_seno" (ver comentários mais adiante)  */
void plota_seno (double, double, double, double, int);

/*==============================================================*\
 *   Função....: main ();                                       *
 *   Parâmetros: nenhum;                                        *
 *   Retorno...: 0 para execução normal e 1 para algum erro;    *
 *   Descrição.: inicializa o modo gráfico e, se  não  houver   *
 *               erro, chama "plota_seno", retornando em  se-   *
 *               guida para o modo texto após o usuário aper-   *
 *               tar alguma tecla.                              *
\*==============================================================*/
int main (void)
{
   /* muda o modo de vídeo para a maior resolução possível */
   if (! _setvideomode (_MAXRESMODE)) {
```

```
      /* caso haja algum erro, imprime uma mensagem de aviso */
      fprintf (stderr, "Graphics error! Press any key to exit...");
      getch (); /* aguarda por uma tecla           */
      exit (1); /* encerra o programa anormalmente */
   }

   /* plota o gráfico do seno de -Pi a Pi usando 100 pontos na interpolação */
   plota_seno (-4 * atan(1), 4 * atan(1), -2, 2, 100);

   getch ();                     /* aguarda por uma tecla  */
   _setvideomode (_DEFAULTMODE); /* retorna para o modo de vídeo padrão */

   return (0);                   /* término normal do programa */
}

/*============================================================*\
 *   Função....: plota_seno ();                               *
 *   Parâmetros: - minx -> início do intervalo a ser plotado;  *
 *               - maxx -> fim do intervalo a ser plotado;     *
 *               - miny -> valor mínimo para o eixo y;         *
 *               - maxy -> valor máximo para o eixo y;         *
 *               - num_pts -> número de pontos;                *
 *   Retorno...: nenhum;                                       *
 *   Descrição.: desenha um retângulo, os eixos (se  necessá-  *
 *               rios), o gráfico do seno, interpolado  line-  *
 *               armente (i.e., por segmentos), e seu título.  *
 *------------------------------------------------------------*
 *   Restrições: - modo gráfico já inicializado;              *
 *               - minx < maxx && miny < maxy;                 *
 *               - num_pts >= 2.                               *
\*============================================================*/
void plota_seno (double minx, double maxx, double miny, double maxy, int num_pts)
{
   /* Macros que transformam as coordenadas do gráfico para a tela */
   /* --->>> Tome cuidado ao utilizar macros desse tipo!!!        */
   #define scr_x(x) (short)(ratio_x * ((x) - minx) + l) /* para abscissas */
   #define scr_y(y) (short)(ratio_y * ((y) - miny) + b) /* para ordenadas */

   int    i;                  /* variável contadora                        */
   short  l, r, b, t;         /* variáveis com as coordenadas da tela      */
   double dx, x,              /* incremento e ponto x para a plotagem      */
          ratio_x, ratio_y;   /* coeficientes de transformação             */
   struct videoconfig vc;     /* configuração do modo de vídeo corrente    */

   /* obtém a configuração do modo de vídeo corrente */
   _getvideoconfig (&vc);

   /* se encontrar as fontes para o texto ... */
   if (_registerfonts (FONPATH) > 0)
      /* ... e a fonte "Roman" estiver disponível ... */
      if (_setfont ("t´roman´h20w15b") >= 0) {
         /* imprime o título do gráfico centralizado */
         _moveto ((vc.numxpixels - _getgtextextent (TITLE)) / 2, (MARGINS-20) / 2);
         _outgtext (TITLE);
```

```
      }

   /* libera a memória alocada por "_registerfonts" */
   _unregisterfonts ();

   /* restringindo a "área de trabalho" */
   _setviewport (MARGINS, MARGINS, vc.numxpixels-MARGINS+1, vc.numypixels-MARGINS+1);

   /* inicialização das variáveis com as coordenadas absolutas */
   /* da tela, configuradas para a nova "área de trabalho"     */
   l = 0;                            /* esquerda */
   t = 0;                            /* acima    */
   r = vc.numxpixels - 2 * MARGINS;   /* direita  */
   b = vc.numypixels - 2 * MARGINS;   /* abaixo   */

   /* inicialização dos coeficientes angulares utilizados na transformação */
   ratio_x =  (r - l) / (maxx - minx);
   ratio_y = -(b - t) / (maxy - miny);

   _rectangle (_GBORDER, l, t, r, b); /* desenhando um retângulo */

   /* desenhando os eixos coordenados se forem necessários */
   if (miny < 0 && maxy > 0) {
      _moveto (l, scr_y (0));
      _lineto (r, scr_y (0));  /* eixo x */
   }
   if (minx < 0 && maxx > 0) {
      _moveto (scr_x (0), t);
      _lineto (scr_x (0), b);  /* eixo y */
   }

   /* inicialização das variáveis utilizadas na plotagem */
   x  = minx;                          /* início do intervalo            */
   dx = (maxx - minx) / (num_pts - 1);   /* p/ distrib. uniform. dos ptos. */

   /* mudando para uma cor diferente, se houver, para plotar o gráfico */
   if (vc.numcolors > 1) _setcolor (2);

   /* plotagem do gráfico seno propriamente dito */
   _moveto (scr_x (x), scr_y (sin(x)));      /* posicionando ponto inicial */
   for (i = 1; i < num_pts; i++) {
      x += dx;                               /* passando para o próx. pto. */
      _lineto (scr_x (x), scr_y (sin(x)));   /* desenhando os segmentos    */
   }

}
```