Métodos e Técnicas de Programação
.:: Avaliação continuada P7 ::.

Prof. Igor Peretta

ENTREGA: até 12/dez/2018

Contents

1 Programas a serem entregues

Os programas a serem entregues precisam seguir o nome da seção em que são descritos, não sendo aceitos programas com outros nomes.

1.1 P7.c

Com base no programa que você criou para o P6, modifique-o para exibir um menu com duas opções:

1. `Gerar arquivo com pontos`
2. `Recuperar pontos de arquivo`

Na opção 1, usando a `struct` "ponto" (coordenadas x e y), peça ao usuário um número N de pontos desejado e grave os pontos gerados em arquivo binário identificado como "pontos.dat".

Na opção 2, usando a `struct` "ponto" (coordenadas x e y), leia o arquivo "pontos.dat" e mostre as N coordenadas dos pontos que dividem uma circunferência de raio um em partes iguais.

Importante: Este programa estende o anterior, portanto você deve usar o programa que você entregou como P6 como ponto de partida. Se seus programas P6 e P7 no repositório tiverem discrepâncias superiores ao esperado com a extensão, o P7 será **desconsiderado**.

1.1.1 Dicas

1. Use `fread()` e `fwrite()`.
2. Você tem duas opções para possibilitar a leitura do binário: (a) ou usa `feof()` e `realloc()`; (b) ou grava primeiro N e depois os pontos. A vantagem da primeira opção é entender laços de leitura (enquanto não for fim de arquivo, realloc a memória); já a vantagem da segunda opção é facilitar a leitura, dividindo-a em duas etapas (primeiro ler N, depois reservar N elementos de "pontos" na memória, depois ler vetor de pontos).
3. Não se esqueça do `getchar()` para limpar o ENTER do buffer de teclado após o `scanf()` na captura das opções do usuário.
4. As bibliotecas a serem usadas: `stdio.h`, `stdlib.h` e `cmath.h`.
5. Não esqueça do `free()` ao final do programa para evitar vazamento de memória.
6. Ao compilar seu código, se estiver tendo problemas com os comandos da biblioteca `math.h`, basta incluir a flag `-lm` no comando de compilação. Exemplo: `gcc -o P7 P7.c -lm`.
7. Para mostrar os pontos, use o especificador `"%.3lf"` para deixar os números com apenas três casas decimais.

1.1.2 Testes

Os seguintes testes devem ser efetuados em sequência a partir de uma nova execução do programa:

- "1" + "7" retorna "Arquivo gravado"
- "2" retorna "(1.000, 0.000) (0.500, 0.866) (-0.500, 0.866) (-1.000, 0.000) (-0.500, -0.866) (0.500, -0.866) (1.000, -0.000)"

2 Informações importantes

É necessário criar em sua conta do github um repositório com o nome 'MTP-2018-2'. É nesse repositório que você dar *upload* do(s) seu(s) código-fonte(s) (ex. arquivos `P1.c`, `P2.c` etc.), não sendo desejado nenhum executável ou arquivo de apoio de projetos.

Em todo programa que você fizer, comece com seu nome e matrícula como comentários. Se não constar essas informações nos arquivos enviados para seu repositório no Github, os programas serão **desconsiderados**.

Mantenha seu código limpo. Não use comandos como `system(pause)` ou `#include<conio.h>` pois são específicos do sistema operacional Windows. Se usá-los, seu código-fonte poderá não compilar, invalidando sua entrega.