Métodos e Técnicas de Programação
.:: Avaliação continuada P5 ::.

Prof. Igor Peretta

ENTREGA: até 07/nov/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 P5.c

Programa para cálculo de soma e de produto de **50 números aleatórios**:

- Copie o trecho abaixo para ter uma função para sortear números pseudo-aleatórios (algoritmo LCG: `https://pt.wikipedia.org/wiki/Geradores_congruentes_lineares`):

```
typedef
    unsigned long long int
Bytes8;
typedef
    struct LCG { Bytes8 a, c, m, rand_max, atual; }
LCG;
```

```
void semente(LCG * r, Bytes8 seed) {
    // constantes do POSIX [de]rand48, glibc [de]rand48[_r]
    // ULL transforma a constante 'int' em 'unsigned long long int'
    r->a = 0x5DEECE66DULL;
    r->c = 11ULL;
    r->m = (1ULL << 48);
    r->rand_max = r->m - 1;
    r->atual = seed;
}
Bytes8 lcg_rand(LCG * r) {
    r->atual = (r->a * r->atual + r->c) % r->m;
    return r->atual;
}
double lcg_rand_01(LCG * r) {
    return ((double) lcg_rand(r))/(r->rand_max);
}
```

- Gere uma função que retorna um vetor de **números aleatórios** entre 0.5 e 1.5:

```
void gera_numeros(float * vetor, int tam,
                  float min, float max, LCG * r) {
    int i;
    for(i = 0; i < tam; i++)
        vetor[i] = (max-min)*lcg_rand_01(r) + min;
}
```

- Não esqueça que a chamada das funções aleatórias deverá ter, como um dos argumentos, o endereço da variável `LCG` criada (ex. `gera_numeros(vetor, N, 0.5, 1.5, &random)`)
- Na primeira linha de `main()`, coloque `LCG random; semente(&random, 123456);` para gerar uma única semente de aleatoriedade (facilidade para correção).
- Com o apoio do material das aulas, implemente a as funções de `float soma(float *inicio_vetor, float *fim_vetor)` e de `float produto(float *inicio_vetor, float *fim_vetor)`.
- Informe o usuário que serão **50 números** aletórios gerados (imprima-os, se quiser).

- Dê ao usuário as opções: `1 - Somatorio` e `2 - Produtorio` .
- Apresente ao usuário o resultado da soma ou do produto dos números aleatórios obtidos, de acordo com a opção desejada.

1.1.1 Dicas

1. Não se esqueça do `getchar()` para limpar o ENTER do buffer de teclado após o `scanf()` na captura da opção do usuário.
2. Fazer a soma e o produto de maneira recursiva.
3. As bibliotecas a serem usadas: `stdio.h`.

1.1.2 Testes

- "1" resulta em "50.725636"
- "2" resulta em "0.273784"

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.