

SME0230 - Introdução à Programação de Computadores

PRIMEIRO SEMESTRE DE 2016

Professora: Marina Andretta (andretta@icmc.usp.br)

Monitores: Amanda Carrijo Viana Figur (amanda.figur@usp.br)
Kleber Roberto Stamboni (kleber.stamboni@usp.br)
Vinicius Volponi Ferreira (vinicius.volponi.ferreira@usp.br)

Exercícios de Laboratório 8

06/05/2016

Data máxima de entrega: 09/05/2016 até às 23h59

Forma de entrega: Os exercícios deverão ser enviados por e-mail para

`exercicios.sme0230.2016@gmail.com`

O assunto do e-mail deverá ser IPC_Lab8. Todos os exercícios devem estar em um único arquivo zip com o seguinte nome IPC_Lab8_<número usp>.

Formato dos arquivos: No início de cada arquivo deve haver um comentário com o nome e o número USP do aluno.

Para cada algoritmo, o nome do arquivo deverá ser `Ex<i>_<número usp>.c` em que <i> representa o número do exercício correspondente.

Exemplo

`Ex1_123456.c`

Observações importantes:

- Trabalhos entregues fora do prazo não serão aceitos.
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. A avaliação dos exercícios levará isto em conta.
- Cada programa deve ser executado tantas vezes quantas forem necessárias para testar todos os casos possíveis para as entradas.

Dica: Para criar um arquivo zip no Linux, basta digitar no terminal

`zip <arquivo de saída>.zip <arquivos de entrada>`

Exemplo

`zip IPC_Lab8_123456.zip Ex1_123456.txt Ex2_123456.c`

Exercício 1

Desenvolva um programa, em linguagem C, que simule contas bancárias com as seguintes funcionalidades:

Cadastro: Cada conta terá um número, nome completo e CPF do cliente e saldo relacionados a ela. Um CPF poderá ser associado a APENAS uma conta. É necessário que seja feito um depósito inicial toda vez que uma nova conta for criada.

Saque e depósito: Um cliente deve ser capaz de efetuar saques ou depósitos em sua conta. Sempre que uma dessas operações for realizada, o sistema deverá apresentar o número da conta, o nome do titular e o saldo final.

Crie um menu principal com as opções:

- (1) `Cadastrar conta`
- (2) `Efetuar saque`
- (3) `Efetuar depósito`
- (4) `Sair do sistema`

O sistema permanecerá em execução até que o comando de sair seja pressionado.

Observações:

- Utilize ponteiros, registros e funções para o desenvolvimento do exercício.
- Ao finalizar o programa, certifique-se que toda memória dinâmica usada foi desalocada.

Exercício 2

Uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais.

Exemplo: A matriz

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$

é um quadrado mágico.

Dada uma matriz quadrada $A_{n \times n}$ em que $2 \leq n \leq 50$, verifique se A é ou não um quadrado mágico.

Observações:

- A matriz deve ser alocada dinamicamente na memória (não se esqueça do `free`!).
- Modularize seu código. Por exemplo: uma função para alocar a matriz na memória, outra para verificar as linhas, etc.