

# Laboratório de Computação I

Prof. Pedro B. Velloso

2º trabalho: Políticos - 2017/1

PRAZO DE ENTREGA: até dia 11 de junho

## Problema

Em um país fictício, chamado Lisarb, estão sendo revelados inúmeros casos de corrupção relacionados a propinas pagas por grandes empresas a políticos. A situação complicou quando duas grandes empresas revelaram listas com nomes de políticos que estariam envolvidos no esquema. O problema é que o número de políticos envolvidos é tão grande que a polícia federal deste país está com dificuldade de organizar os dados. Basicamente, eles precisam armazenar os nomes dos políticos de cada partido e marcar quem é suspeito de ter recebido propina. Portanto, eles gostariam de armazenar os políticos de um partido em um arquivo que conteria o nome de cada um deles, a idade, e a informação se ele aparece em alguma das listas. Visto que são duas listas principais de denúncias, a PF gostaria de gerar relatórios com a lista de políticos que não estão sob suspeita, de políticos que aparecem na lista da empresa Thcerbedo, políticos que aparecem na lista da empresa SBJ e políticos que aparecem na duas listas.

## Objetivo

O trabalho consiste em fazer um programa em C que implemente um sistema para armazenar as informações dos políticos do Lisarb e que permita a consulta dos dados.

## Características básicas

O sistema deve armazenar as seguintes informações dos políticos, em um registro (*structure*): **nome**, **idade** e **suspeita**. Os políticos serão agrupados por partidos, de maneira que cada arquivo do sistema contenha políticos de um único partido. A **suspeita** dos políticos será representada por um caracter apenas, onde:

- ‘0’ – político não aparece em nenhuma das listas;
- ‘1’ – político aparece apenas na lista da empresa Thcerbedo;
- ‘2’ – político aparece apenas na lista da empresa SBJ;
- ‘3’ – político aparece em ambas as listas;

Para acessar a lista de políticos de um determinado partido, o usuário deve antes de tudo abrir o arquivo. Como o usuário pode abrir vários arquivos ao mesmo tempo, o sistema terá uma lista simplesmente encadeada de arquivos abertos. Esta lista conterà o nome do arquivo aberto, o nome do partido associado a ele e o ponteiro para o arquivo (*FILE \* fp*). A lista de arquivos abertos deve ser mostrada a todo momento no alto do menu do programa. Assim, todas as funcionalidades do sistema só poderão ter efeito no partidos cujos arquivos já foram abertos. A lista de arquivos abertos deve ser ordenada pelo nome do partido.

## Funcionalidades

1. abrir um arquivo de um partido;
2. fechar um arquivo de um partido;
3. cadastrar político;
4. apagar político pelo nome;
5. mostrar todos os políticos de um partido, em forma de tabela;
6. mostrar todos os políticos, em forma de tabela (nome, id, partido), que:
  - (a) não estão sob suspeita;
  - (b) aparecem em apenas uma lista (mostrar a empresa após a idade);
  - (c) aparecem em ambas as listas.

A inclusão de um político no arquivo do seu partido deve ser feita de forma ordenada.

## Programa principal

- deve conter um menu com as opções das operações e só deve terminar a execução quando o usuário escolher a opção de sair.
- não deve conter variáveis globais;
- o programa deve ser modular, ou seja, deve chamar as funções e os procedimentos implementados;

## Limitações

- o programa não deve conter vetores nem matrizes.

## Dicas

- implementar o programa de acordo com as especificações dadas, pois cada item fora da especificação acarretará a perda de pontos;
- outras funcionalidades que não foram especificadas que ajudem e facilitem a utilização do programa serão compensadas com pontos extras;
- fazer passo à passo do que é pedido, sempre compilando e testando cada parte implementada.

## Avaliação

Os programas serão avaliados na aula de laboratório. O trabalho é individual e qualquer tipo de cola será punida com ZERO. O aluno deve compilar o trabalho na hora. O aluno deverá saber explicar o código na hora da apresentação. Além do bom funcionamento do programa, será avaliado também:

- programas devem ser comentados;
- indentação adequada dos programas;
- nomes intuitivos de variáveis;
- interface com o usuário - facilidade de uso;
- entendimento do próprio programa;
- modularidade.

## Entrega

O trabalho deve ser enviado ao e-mail do professor de laboratório até o dia 11 de junho, às 23h55. Trabalhos entregues após esta data serão desconsiderados. O aluno deve colocar o nome do arquivo *.c* com o seu nome, exemplo: *pedro – veloso.c*. Este arquivo deve estar dentro de um diretório com o nome do aluno, exemplo: *pedro – veloso*. Este diretório deve ser zipado e enviado para o e-mail especificado. O diretório DEVE conter apenas o arquivo *.c*.

Emails para entrega dos trabalhos:

EL1 - Pedro B. Velloso – *veloso@poli.ufrj.br*

EL2 - Pedro Cruz – *cruz@gta.ufrj.br*

EL3 - Carlos Henrique – *choma@gta.ufrj.br*