



## Trabalho 2 de LP1 21/11/2016

O Trabalho 2 consiste em melhorar o programa pedido no trabalho anterior. Conteúdos teóricos que podem ser explorados para a melhoria do código do programa entregue no Trabalho 1 são:

- *Structs* e vetores de *Structs*, na realização das votações, **implementação obrigatória**;
- Funções, desenvolvimento **obrigatório**;
- Bibliotecas, desenvolvimento **obrigatório**;
- Ponteiro e alocação dinâmica (opcional, contudo se **implementada corretamente**, terá melhora na nota final do trabalho T2);
- Arquivos, implementação **obrigatória**;
- Etc...

No programa principal deve ser somente implementados os menus, sendo que cada opção do Menu 2 deve chamar uma função para executar aquilo que se pede.

- Inicialmente há um menu com duas opções:

### Menu 1

Escolha sua opção

0 – Iniciar o programa

1 – Finalizar o programa

- Se o usuário escolher “Finalizar o programa”, o programa termina. Caso contrário, é apresentado o seguinte menu:

### Menu 2

Escolha o tipo de votação

0 – Votação secreta

1 – Votação aberta

2 – Voltar ao menu anterior

Ao se escolher a opção “0” (votação secreta), o programa deve chamar uma função que recebe um valor *n* que representa o número total de votantes e a partir daí apresentar *n* vezes um as opções “Sim”, “Não” e “Encerrar a votação”.

Ao final da votação, deve ser apresentado na tela quantos foram os votos para “Sim”, quantos votos foram para “Não” e quantas foram as abstenções (abstenções são contabilizadas quando a votação é encerrada e não foram



computados todos os votos possíveis). O programa deve também utilizar uma função para dizer se a votação foi válida (pelo menos 50% + 1 do total de votantes votaram) e se houve maioria para o “Sim” ou para o “Não” (se uma das opções recebeu 50% + 1 de votos dentre aqueles que votaram). Todos os resultados devem ser também escritos em um arquivo e deve ser gerado um arquivo de resultado por votação.

Ao se escolher a opção “1” (votação aberta), o programa deve chamar uma função que recebe um valor *n* que representa o número total de votantes e um vetor de “parlamentares” (a *struct* “Parlamentar” deve ser definida). A partir daí receber *n* vezes os dados do parlamentar e sua opção de voto. Os dados da *struct* parlamentar devem armazenar:

- Código do parlamentar;
- Partido;
- Estado;
- Sexo;

Após isso, deve ser contabilizado o voto, sendo as seguintes opções: “Sim”, “Não” e “Abstenção”.

No final, devem ser chamadas funções obrigatoriamente para apresentar os seguintes resultados:

- Número total de votos para o “Sim”, “Não” e abstenções e se houve maioria (se uma das opções recebeu 50% + 1 de votos dentre aqueles que votaram);
- Número de votos para “Sim”, “Não” e abstenções por partido;
- Número de votos para “Sim”, “Não” e abstenções por Estado;
- Número de votos para “Sim”, “Não” e abstenções por sexo.

Exemplo do resultado dos votos abertos

Votos totais

Sim	10	62,50%
Não	1	6,25%
Abstenções	5	31,25%
	16	100%

Votos por partido

PVC – Partido Vira Casaca

Sim	1	100%
Não	0	0%
Abstenções	0	0%



1 100%

Votos por estado

São Paulo

Sim 1 100%

Não 0 0%

Abstenções 0 0%

1 100%

Etc...

Todos os resultados devem ser escritos também em um arquivo e deve ser gerado um arquivo de resultado por votação. Após apresentar o resultado, o programa termina.

ATENÇÃO!! TODA entrada de dados no programa deve ser devidamente tratada!

Exemplos: o Menu 1 só deve aceitar os valores 0 ou 1, o Menu 2 só deve aceitar os valores de 0 a 3, etc.

Para o trabalho, considere que:

- Há somente cinco partidos válidos (PU – Partido da União, PPT – Partido Planeta Terra, PVC – Partido Vira Casaca, PdP – Partido do Povo e PPP – Partido Partidário Público). Estes nomes de partidos devem ser recuperados de um arquivo previamente gerado. Leitura de arquivo. Lembre-se que a lista de partidos pode ser alterada a qualquer momento, portanto, seu programa deve se adequar ao novo conteúdo do arquivo com os partidos.

- O código do Parlamentar é um valor de 0 a 600;



### **Componentes por grupo:**

No máximo, 02 membros por grupo.

Restrição: os dois membros do grupo devem fazer parte da mesma turma (A, B ou C).

Entrega:

- Data da entrega: 01/12/2016
- Entregar o código fonte via Moodle. Os membros do grupo devem estar preparados para responder qualquer pergunta sobre o programa.
- Entregas atrasadas sofrerão um decréscimo de 50% na nota por dia de atraso.

### **Critérios de Avaliação**

- Documentação do código (comentários no código fonte);
- Execução correta;
- Organização (utilização das regras para declaração de variáveis e constantes, indentação, etc);
- A identificação de trabalhos copiados fará com que as duplas envolvidas tenham a nota ZERADA.

Dica:

Antes de começar a codificar, planeje o funcionamento dos menus do programa e as funcionalidades em cada opção quando escolhida.