



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO – DEC

DISCIPLINA: LINGUAGENS DE PROGRAMAÇÃO 2
PROFESSOR ANTONIO CARLOS SOBIERANSKI
a.sobieranski@ufsc.br

ENUNCIADO TRABALHO T2 – PERSON

A ser desenvolvido em: DUPLAS

O trabalho T2 consiste na implementação de um programa para cadastro de pessoas físicas e jurídicas. O mesmo deve manter a persistência dos dados armazenados em disco em um arquivo chamada “**database.dat**”, o qual conterá todas as informações conforme a execução do menu abaixo ocorrer:

MY PAPA'S COMPANY - SIG

1. Add PF...
2. Add PJ...
3. Remove PF Index...
4. Remove PJ Index...
5. Print Names in Ascending Order
6. Exit

Enter an option:

Menu 1 e 2. As opções 1 e 2 realizam a inserção de uma nova pessoa física ou jurídica, cada um dos campos seguindo a sequência definida na struct abaixo:

```
struct personPF
{
    string nome;
    string nomeMae;
    string cpf;
    string endereco;
    string telefone;
};

struct personPJ
{
    string razaoSocial;
    string cnpj;
    string endereco;
    string telefone;
    size_t capitalSocial;
};
```

Usar *cout's* solicitando cada um dos campos, de forma livre (*opcional fdebug*)

Menu 3 e 4. As opções 3 e 4 removem, respectivamente, uma pessoa física ou jurídica pelo seu índice. Aqui deve se preocupar em o usuário não digitar um índice fora do tamanho do vector. Se for esse o caso, imprimir mensagem na tela de que índice não pode ser removido, e retorna ao menu principal.

5. A opção 5 imprime todos os dados em tela em ordem alfabética. Deve ser impresso somente nomes seguido por quebra de linha. Não precisa diferenciar entre PF e PJ.

6. Sair do sistema.

IMPORTANTE: Para o presente entregável T2, serão avaliados 2 aspectos:

* A saída em tela do menu 5, que é a única que deve ser mostrada (independente de ndebug)

* o arquivo *database.dat* final

O programa deve ler o arquivo *database.dat* a cada nova execução do programa, carregando todas as informações na memória corretamente. Caso o arquivo não exista em uma primeira execução do programa, um novo *database.dat* deve ser criado. A cada nova inserção ou remoção, o arquivo deve ser atualizado.

A ordem que o arquivo dataset será salvo será a seguinte:

* Primeiro, todas as pessoas físicas, seguindo a ordem do vetor, cada um precedido por código 1 (ver video exemplo, ou arquivo *database.dat* mais abaixo)

* Segundo, todas as pessoas jurídicas, seguindo a ordem do vetor, cada um precedido por código 2 (ver video exemplo, ou arquivo *database.dat* mais abaixo)

Obs: o arquivo "database.dat" é bastante simples, apresentando na sequencia todas as informações demonstradas na **struct** mostrada na figura anterior, **PRECEDIDO** por código 1 para pessoa física, e código 2 para jurídica, conforme exemplo de saída abaixo. O arquivo "database.dat" sempre possui uma quebra de linha ao seu final.

Inputs:	Saída em Tela	Saída no arquivo <i>database.dat</i>
Primeira execução usando input1.txt , sem database.dat		
1 Antonio Sobieranski Mamae 000.000.000-77 UFxC 48 8888 8888 1 Jose da Silva Joseph's Mama 222.222.222-33 None 48 9999 9999 2 MaxPrint Solucoes de Impressao 00000000000000 None 48 0000 1111 15000 1 Pedro Pereira Pedro's Mama 444.444.444-44 None None 5 6	Names in Ascending Order: Antonio Sobieranski Jose da Silva MaxPrint Solucoes de Impressao Pedro Pereira	1 Antonio Sobieranski Mamae 000.000.000-77 UFxC 48 8888 8888 1 Jose da Silva Joseph's Mama 222.222.222-33 None 48 9999 9999 1 Pedro Pereira Pedro's Mama 444.444.444-44 None None 2 MaxPrint Solucoes de Impressao 00000000000000 None 48 0000 1111 15000
Segunda execução usando input2.txt , com database.dat criado anteriormente		
2 Sorvetes LALAH 00000000000000 None 48 0000 1111 10000 2 Acai do Pedrao 00000000000000 None	Names in Ascending Order: Acai do Pedrao Andre Pereira da Silva Jose da Silva Pedro Pereira Sorvetes LALAH	1 Jose da Silva Joseph's Mama 222.222.222-33 None 48 9999 9999 1 Pedro Pereira Pedro's Mama 444.444.444-44

48 0000 3333 3500 1 Andre Pereira da Silva Mamas Boy 111.111.111-33 Rua das Amoras, 33 48 8888 9999 3 0 4 0 5 6		None None 1 Andre Pereira da Silva Mamas Boy 111.111.111-33 Rua das Amoras, 33 48 8888 9999 2 Sorvetes LALAH 0000000000000 None 48 0000 1111 10000 2 Acai do Pedrao 0000000000000 None 48 0000 3333 3500
--	--	---

Para agilizar o teste do algoritmo, utilizar redirecionamento (opcionalmente):

Linux: `./executavel < input1.txt`
Windows: `executavel < input1.txt`

Dicas: uso de *structs* em *std::vector*, *ifstream* para ler, *ofstream* para gravar arquivo. Usar *getline* usando o *cin* como input, para ler strings com espaços.

Alguém disse: Debug salva vidas....