



Centro de Ciências Exatas, Arquitetura e Engenharia

Professor:	Ciro Cirne Trindade
Disciplina:	Estruturas de Dados-II
Cursos:	Ciência da Computação/Sistema de Informação – 4º semestre
Grupos:	3 alunos
Data de Entrega:	14/10/2019

1º Trabalho Prático – Rede Social

Uma rede social armazena as informações a respeito de seus usuários em uma **árvore AVL**. As informações de cada usuário da rede social são: seu nome de usuário (identificador único), nome completo, sexo e sua lista de amigos.

A estrutura de cada nó da árvore AVL deve ser a mostrada abaixo:

```
typedef struct {
    char usuario[21]; // nome de usuário
    char nome_completo[51]; // nome completo
    char sexo;
    lista amigos; // lista encadeada ordenada dos amigos
} avl_info;

typedef struct avl_no * avl_tree;

struct avl_no {
    avl_info dado;
    avl_tree esq;
    avl_tree dir;
    int bal; // balanceamento do nó
};
```

Escreva um programa em C que modele essa rede social e tenha as seguintes funcionalidades:

- a) Carregar usuários: o sistema deve pedir o nome do arquivo onde as informações da rede estão cadastradas e chamar a função descrita a seguir:

- `void load_users(avl_tree * t, char * filename):` a configuração inicial da rede social deve ser carregada a partir de um arquivo texto (`filename`) onde cada usuário é descrito por duas linhas. A primeira linha contém o nome de usuário, seu nome completo e sexo ('m' ou 'f'), separados por um ; (ponto e vírgula). A segunda linha contém os nomes de usuários de seus amigos, também separados por um ; (ponto e vírgula). Abaixo é mostrado um exemplo deste arquivo com 3 usuários. *Dica: use a função `strtok()` para quebrar uma string em pedaços.*

```
joao;Joao da Silva;m
pedro; maria
```

```

pedro;Pedro Barbosa;m
joao
maria;Maria do Carmo;f
joao

```

- b) Adicionar um novo usuário: o sistema deve solicitar os dados do novo usuário (nome de usuário, nome completo e sexo) e chamar a função descrita a seguir:
- `bool add_new_user(avl_tree * t, avl_info x):` acrescenta um novo usuário à rede social. A lista de amigos do usuário deve ser inicialmente vazia. Para que a operação seja realizada, o nome de usuário deve ser único, ou seja, não pode haver nenhum usuário cadastrado com esse nome. A função devolve `true` para indicar o sucesso da operação; ou `false`, para indicar a falha.
- c) Adicionar um amigo: o sistema deve solicitar o nome de usuário do usuário e do amigo e chamar a função descrita a seguir:
- `bool add_friend(avl_tree t, char * username, char * friendname):` acrescenta um amigo (`friendname`) a lista de amigos de um usuário (`username`). Tanto o usuário quanto seu amigo já devem estar cadastrados na rede social para que a operação seja realizada com sucesso e `username` deve ser diferente de `friendname`. A função devolve `true` para indicar o sucesso da operação; ou `false`, para indicar uma falha;
- d) Listar usuários: o sistema de listar todos os usuários da rede através da função descrita a seguir:
- `void list_users(avl_tree t):` gera uma listagem em ordem alfabética pelo nome de usuário de todos os usuários da rede (imprime o nome de usuário, nome completo e sexo);
- e) Listar amigos: o sistema deve solicitar um nome de usuário e listar todos os seus amigos através da função descrita a seguir:
- `bool list_friends(avl_tree t, char * username):` mostra o nome de usuário, nome completo e sexo de todos os amigos de um usuário (`username`). A função devolve `true` se o usuário (`username`) estiver cadastrado na rede; ou `false`, caso contrário;
- f) Remover usuário: o sistema deve solicitar o nome de usuário do usuário que deve ser removido e o sistema deve realizar a operação através da função descrita a seguir:
- `bool remove_user(avl_tree * t, char * username):` remove o usuário (`username`) da rede social e também da lista de amigos dos demais usuários. A função deve verificar se o usuário existe para fazer a operação e deve devolver `true` para indicar seu sucesso; ou `false`, para indicar a falha.
- g) Finalizar o programa: o sistema de atualizar o arquivo texto que contém a descrição da rede social (o mesmo utilizado pela função `load_users()`) através da função descrita a seguir:
- `void save_users(avl_tree t):` grava em um arquivo texto as informações da rede, no mesmo formato descrito no item a).

Informações importantes sobre o trabalho

1. Critérios de avaliação:
 - a) Corretude: 70%
 - b) Interface: 20%
 - c) Legibilidade: 10%

2. Todos os trabalhos devem possuir no cabeçalho a identificação dos autores.
3. Trabalhos copiados (com ou sem eventuais disfarces) terão nota dividida pelo número de cópias (inclusive o original).
4. Trabalhos atrasados terão um desconto de 1 ponto por dia atrasado até o limite de 5 dias.
5. Trabalhos com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
6. 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 legibilidade do código será levada em consideração pelo critério de avaliação do trabalho.
7. Você deve enviar os arquivos compactados através do link específico no Moodle.