Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Informação (INF02827)
1º Trabalho Prático[1]
Período: 2009/1
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Data de Entrega: 23/05/2009 ***Grupos de 2 pessoas***

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Lista.

Regras Importantes

- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Para cada dia de atraso, é retirado um ponto da nota do trabalho.

Material a entregar

- Impresso: Documentação do trabalho, que deve conter:
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  - Implementação: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa como discutido em sala de aula.
  - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

- Por email (pdcosta@inf.ufes.br):
  - O assunto da mensagem deve ser ei20091:trab1:<nome1>:<nome2>
    - Por exemplo: ei20091:trab1:<joaosilva>:<mariacosta>
  - Documentação do trabalho (em formato PDF).
  - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
  - O makefile.
  - Favor nomear os arquivos da seguinte maneira: TadListaPessoas.h, TadListaPessoas.c, TadListaAmigos.h, TadListaAmigos.c, e MeuOrkut.c.

Implementando o seu Orkut

Você foi contratado pela Google para refazer o Orkut. A sua primeira tarefa é implementar um programa para controlar as relações de amizade dentro do Orkut. De forma geral, para cada pessoa cadastrada você tem que armazenar a sua lista de amigos. Uma forma de visualizar os relacionamentos é através de uma estrutura chamada *grafo*. Um grafo é composto por vários nodos que são conectados por arestas indicando algum tipo de relação entre esses nodos. Um grafo pode ser implementado através de listas encadeadas: basicamente, implementa-se uma lista onde cada registro contém os dados de um nodo. Além disso, cada registro desta

[1] Trabalho baseado no material do Prof. Luiz Chaimowicz (DCC-UFMG) , usado com permissão

lista contém uma lista encadeada indicando quais os nodos que estão ligados a ele. Isso pode ser observado no seguinte exemplo: Joao é amigo da Maria, do Jose e do Carlos. Maria não tem amigos além do Joao. Carlos é amigo de Joao, Jose e Pedro. Alice não é amiga de ninguém. A figura abaixo mostra o grafo e a estrutura de dados correspondente. Note que cada pessoa tem um identificador único dentro do Orkut (campo chave ID), e esse campo é utilizado para construir as listas de amigos:

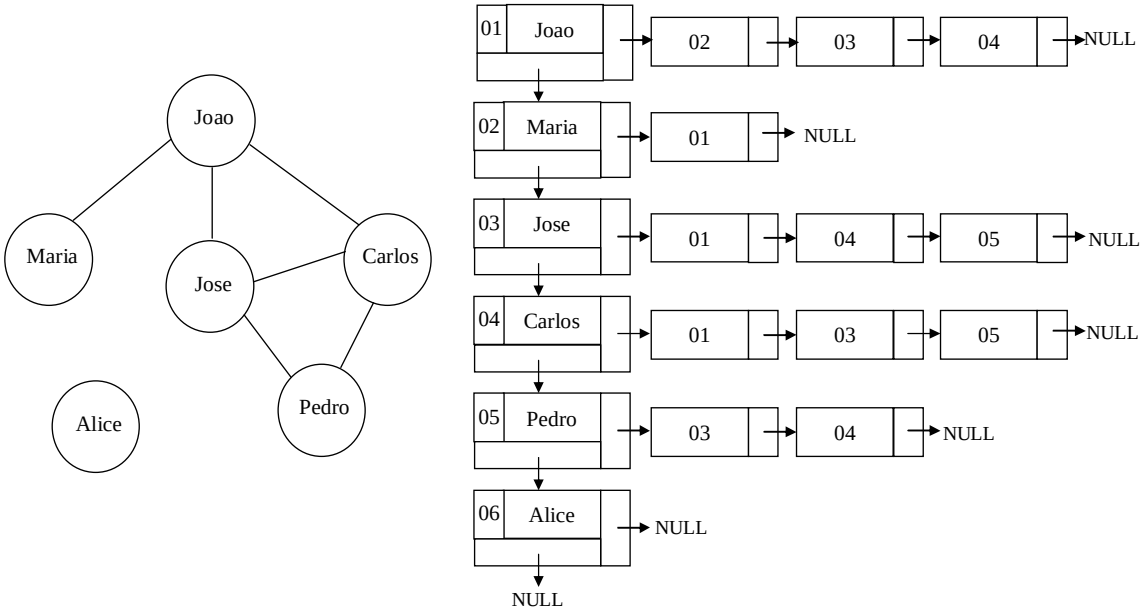

Figura 1 - Orkut

Nesse trabalho, você deverá implementar essa estrutura. Considere que cada pessoa é um registro contendo os campos nome, identificador único e a cidade onde mora a pessoa, e que não existem homônimos no Orkut. Considere somente nomes simples no Orkut, tanto para cidade quanto para pessoas. As seguintes operações devem ser implementadas:

- Cadastra(Pessoa): cadastra uma nova pessoa no Orkut. O identificador único dessa pessoa deve ser gerado automaticamente.
- Remove(Nome): remove a pessoa identificada do Orkut. Note que todas as relações de amizade dessa pessoa devem ser removidas.
- Amigos(Nome1, Nome2): Cria uma relação de amizade entre duas pessoas já cadastradas.
- Briga(Nome1, Nome2): Remove uma relação de amizade existente.
- Frequencia(Cidade): Imprime o número de pessoas cadastradas de uma determinada cidade.
- Caminho(Nome1, Nome2): Verifica se existe um caminho de amizade entre Nome1 e Nome2 (caminho direto ou indireto). Retorna SIM caso exista e NAO caso contrário. Por exemplo, Caminho(Joao, Pedro) retorna SIM e Caminho(Joao, Alice) retorna NAO;
- ImprimeDados(Nome): Imprime os dados de uma pessoa (nome, identificador, cidade) bem como o nome, o identificador e a cidade de todos os seus amigos.
- ImprimeTudo(): imprime os dados de todas as pessoas cadastradas bem como as listas de amigos (com nome e cidade de cada um).

A implementação da estrutura deverá ser feita utilizando alocação dinâmica de memória (ponteiros). Você deve fazer testes de consistência se essas operações podem ser aplicadas (teste de pré condições) e deve imprimir mensagens de sucesso ou falha (por exemplo, a chamada de função *Remove(Joana)* no exemplo acima deve dar uma mensagem de erro pois não há ninguém cadastrado com esse nome). Além disso, procure escrever funções auxiliares que facilitem a implementação das operações acima, evitando a repetição

desnecessária de código. Por exemplo, funções de manipulação de listas, pesquisa de id por nome, etc. Organize o seu sistema usando o conceito de **Tipos Abstratos de Dados** discutido em sala de aula.

Entrada e Saída

O seu programa (MeuOrkut) deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (Faz parte do trabalho descobrir como manipular arquivos e strings em C). Exemplo de execução do programa a partir da linha de comando:

```
MeuOrkut entrada.txt
```

O arquivo de entrada é basicamente uma lista de comandos (um por linha) em formato texto. O último comando é a palavra FIM, que indica o final do arquivo. O formato a ser usado é exemplificado abaixo:

Exemplo de arquivo de entrada (entrada.txt)

```
CADASTRA nome cidade (ex. CADASTRA Maria Vitoria)
CADASTRA nome cidade (ex. CADASTRA Jose Guarapari)
CADASTRA nome cidade (ex. CADASTRA Joao Guarapari)
REMOVE nome (ex. REMOVE Rafael)
AMIGOS nome1 nome2 (ex. AMIGOS Maria Jose)
AMIGOS nome1 nome2 (ex. AMIGOS Maria Joao)
AMIGOS nome1 nome2 (ex. AMIGOS Jose Joao)
BRIGA nome1 nome2 (ex. BRIGA Maria Jose)
FREQUENCIA cidade (ex. FREQUENCIA Guarapari)
IMPRIME_DADOS nome (ex. IMPRIME_DADOS Maria)
CAMINHO nome1 nome2 (ex. CAMINHO Jose Alice)
CAMINHO nome1 nome2 (ex. CAMINHO Maria Jose)
IMPRIME_TUDO (ex. IMPRIME_TUDO)
FIM
```

Os comandos FREQUENCIA, CAMINHO, IMPRIME_DADOS e IMPRIME_TUDO deverão imprimir as informações em um arquivos de saída, a ser criado pelo seu programa. O nome do arquivo de saída deverá ser *saida.txt*. Todas as mensagens de erro também deverão ser impressas no arquivo saida.txt.

O comando FREQUENCIA imprime em uma linha o número de pessoas cadastradas de uma determinada cidade. O comando CAMINHO imprime em uma linha SIM se for houver um caminho de amizade entre nome1 e nome2 e NAO caso contrário.

O comando IMPRIME_DADOS (nome) deverá imprimir em uma linha as informações daquela pessoa (nome, identificador e cidade), separados por espaço. Todos os amigos desta pessoa serão impressos nas linhas subsequentes (um por linha), cada linha contendo o nome, o identificador e a cidade da pessoa. Antes das informações dos amigos, inclua uma linha com a string “Amigos de ” + nome da pessoa. Depois de impressa as informações dos amigos, imprima uma linha em branco. Considerando o exemplo na Figura 1, a sequência de comandos IMPRIME_DADOS (Maria), IMPRIME_DADOS (Pedro), imprime as seguintes linhas no aquivo saida.txt:

```
Maria 02 Vitoria
Amigos de Maria
Joao 01 Guarapari

Pedro 05 Serra
```

```
Amigos de Pedro
Jose 03 Guarapari
Carlos 04 Vitoria
```

O comando IMPRIME_TUDO deverá imprimir as informações de todas as pessoas do Orkut usando o mesmo formato de saída da função IMPRIME_DADOS (nome).

Considerando o arquivo de entrada dado acima, e que são gerados pelo programa identificadores únicos para os usuários, espera-se o seguinte no arquivo de saída:

Arquivo de saída para o arquivo entrada.txt

```
Erro: Rafael não esta cadastrado no Orkut

2

Maria 01 Vitoria
Amigos de Maria
Joao 03 Guarapari

Erro: Alice não esta cadastrado no Orkut

SIM

Maria 01 Vitoria
Amigos de Maria
Joao 03 Guarapari

Jose 02 Guarapari
Amigos de Jose
Joao 03 Guarapari

Joao 03 Guarapari
Amigos de Joao
Maria 01 Vitoria
Jose 02 Guarapari
```

BOM TRABALHO!