Universidade Federal do Ceará
Departamento de Computação
Curso de Ciência da Computação

Disciplina	Programação - CK0226	**Semestre**	2017/2
Professor	Lincoln Souza Rocha		
Trabalho Final da Disciplina			

INSTRUÇÕES

Prazo de Entrega: 18 de dezembro de 2017, até 23:59. Upload através do. Não serão aceitas entregas por e-mail (a não ser que haja algum problema com a submissão pelo SIGAA). Os trabalhos será avaliados com base na análise do código fonte, da execução e de arguição, como descrito as seguir. Após a data de entrega, será informada uma lista de alunos, representando uma amostragem de 25% da turma, para arguição individual.

DESCRIÇÃO GERAL

O objetivo principal deste trabalho é implementar uma rede social de compartilhamento de agenda de viagens entre amigos. Nessa perspectiva, os usuários possuem um identificador único, um nome, e uma lista de amigos, e uma lista de agendamentos de viagens contendo informações sobre a data da viagem (dia, mês e ano), local de destino (nome da cidade e do país) e período de permanência no local de destino (quantidade de dias). Usuários que são amigos (isto é, usuários que estão, simultaneamente, na lista de amigos cada um), podem consultar a agenda de viagens uns dos outros para verificar possibilidades de encontros ocasionais durante suas viagens ou mesmo fazer uma planejamento conjunto afim de fazerem viagens juntos. Além disso, usuários podem cadastrar, buscar (listar todas ou filtrar por data de viagem e local de destino), alterar e remover agendamentos de viagens, fazer consultas sobre agendamento de viagens de amigos (listar todas ou filtrar por data de viagem e local de destino). Além disso, a rede social permite que o usuário cadastre, liste e remova amigos.

DETALHES DE IMPLEMENTAÇÃO

O trabalho deve seguir as seguintes restrições de implementação. Os usuários e agendamento de viagens devem ser implementados como TAD (Tipo Abstrato de Dados).

```
typedef struct viagem Viagem;

Viagem *nova_v(int dia, int mes, int ano, char *cidade, char *pais, int periodo);
void libera_v(Viagem *viagem);
void acessa_v(Viagem *viagem, int *dia, int *mes, int *ano, char *cidade, char *pais, int *periodo);
void atribui_v(Viagem *viagem, int dia, int mes, int ano, char *cidade, char *pais, int periodo);
void atribui_direita_v(Viagem *viagem, Viagem *direita);
Viagem *acessa_direita_v(Viagem *viagem);
void atribui_esquerda_v(Viagem *viagem, Viagem *esquerda);
Viagem *acessa_esquerda_v(Viagem *viagem);
int tamanho_v();
```

```
#include “Viagem.h”
typedef struct usuario Usuario;

Usuario *novo_u(int id, char *nome);
void libera_u(Usuario *usuario);
void acessa_u(Usuario *usuario, int *id, char *nome);
void atribui_u(Usuario *usuario, int id, char *nome);
void adiciona_amigo_u(Usuario *usuario, Usuario *amigo);
void remove_amigo_u(Usuario *usuario, int id);
Usuario *busca_amigo_u(Usuario *usuario, int id);
Usuario *lista_amigos_u(Usuario *usuario);
void adiciona_viagem_u(Usuario *usuario, Viagem *viagem);
void remover_viagem_u(Usuario *usuario, int id);
Viagem *listar_viagens_u(Usuario *usuario);
Viagem *buscar_viagem_por_data_u(Usuario *usuario, int dia, int mes, int ano);
Viagem *buscar_viagem_por_destino_u(Usuario *usuario, char *cidade, char *pais);
Viagem *filtrar_viagens_amigos_por_data_u(Usuario *usuario, int dia, int mes, int ano);
Viagem *filtrar_viagens_amigos_por_destino_u(Usuario *usuario, char *cidade, char *pais);
Usuario *filtrar_amigos_por_data_viagem_u(Usuario *usuario, int dia, int mes, int ano);
Usuario *filtrar_amigos_por_destino_viagem_u(Usuario *usuario, char *cidade, char *pais);
int tamanho_u();
```

Sobre a descrição dos arquivos de cabeçalho Viagem.h e Usuario.h:

- Campos que compõem a estrutura da viagem: int id, int dia, int mes, int ano, char *cidade (tamanho 60), char *pais (tamanho 30), int periodo, Viagem *direita e Viagem *esquerda;
- Campos que compõem a estrutura usuário: int id, char *nome (tamanho 80), Usuario **amigos (vetor de amigos) e Viagem *viagens (lista de agendamento de viagens);
- A adição de amigos deve ser feita ao seguindo o modelo de fila, porém, a remoção deve ser aleatória dependendo do id informado (lembre-se de reajustar o vetor após remoções);
- A adição de viagens deve ser implementada como Árvore Binária de Busca, garantindo a ordem pela data da viagem;
- Agendamentos não podem possuir mesmos ids e nem serem sobrepostos, isto é, não é possível cadastrar um novo agendamento de viagem cujo o período total de permanência, contado à partir da data da viagem, se sobreponha com um agendamento previamente cadastrado.