Universidade Federal de São Carlos
Estrutura de Dados I
Trabalho Final

A cada ano novos alunos vão ingressando na Universidade Falho Sistema Carregado (UFSCar), e com isso surgiu a necessidade de implementar um sistema mais eficiente que o atual (implementado com lista simples) devido ao grande tempo de espera no manuseio de dados. O programa consiste básicamente em gerenciar os dados relacionados com os alunos (Nome, RA, IRA e Status da Matrícula).

Sabemos que você consegue deixar esse sistema mais eficiente através do emprego de árvores binárias. Utilize seu conhecimento para ajudar a instituição!

Utilize duas árvores binárias em memória, uma ordenada por RA e outra por Nome do aluno. As árvores deverão ter o mesmo número de nós, mas não necessariamente nos mesmos ramos da árvore. Você pode utilizar a árvore balanceada para melhorar o desempenho da sua busca.

O seu programa deverá ser capaz de:

- **Cadastrar novos alunos:** O usuário irá informar o RA, nome (de até 64 caracteres), IRA (de 0 até 99999) e Status da Matrícula (1 ou 0). Salve sempre o nome do aluno em caixa alta.
- **Realizar buscas:** Busca limpa (apenas alunos ativos) relativa por nome, ou seja caso o usuário digite “Mar” você deve listar todos os alunos que começem com “MAR”; O mesmo para busca suja, com a diferença de que nessa se lista também os alunos desativados e por fim busca por RA. Para todas as buscas, caso não haja nenhum aluno dentro dos parâmetros informados imprima a mensagem de aluno não encotrado.
- **Alterar o IRA dos alunos** através do RA. Caso o aluno exista, altere o IRA informado pelo usuário e imprima o `define` de alterado com sucesso. Caso não seja encontrado imprima o `define` de aluno não encontrado.
- **Alterar o estado de um aluno** (ativo ou inativo): Através do RA e do novo estado informado, caso o aluno não seja encontrado, imprima o `define` do aluno não encontrado.
- **Listar –** Permite que o usuário liste todos os alunos ativos ou não em ordem alfabética.
- **Salvar os dados em memória para o disco:** Um arquivo para cada árvore, nome.dat e ra.dat.
- **Recuperar informações do disco para a memória:** verifique se os arquivos existem, caso sim, carregue-os para a memória primária.
- **Liberar a memória e encerrar**: Toda a memória primária alocada deve ser liberada antes do encerramento do programa.

Observações:

Todas as informação de saída já foram estabelecida com o uso de `defines`, utilize apenas as constantes para imprimir informações na tela.

É necessário verificar se o cadastro de um novo aluno não irá gerar RA duplicado no sistema, se sim, não cadastre e imprima o define desse erro. Considere que não existirão nomes idênticos.

É recomendado utilizar as estruturas já fornecidas nos arquivos do classroom. Porém alterações são permitidas desde que comentadas explicitamente no `header` e a estrutura continue respeitando a definição de uma árvore binária de busca.

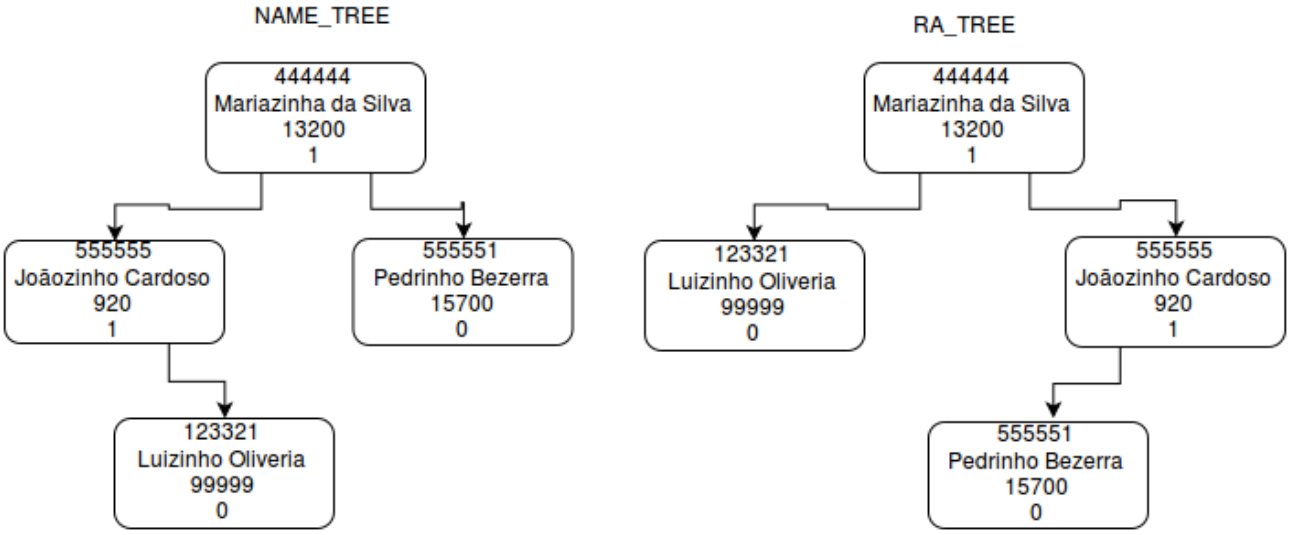
Não se preocupe com a duplicação de dados na memória. Técnicas de indexação serão vistas futuramente.

Dicas:

- Busca relativa pode ser fácilmente implementada utilizando a função strncmp. Desde que você trate todas as strings de entrada para o mesmo padrão de caixa alta.
- Utilize o conceito de travessia pré-ordem para salvar a árvore no arquivo.
- Evite o uso de funções como fflush e __fpurge, de preferência à técnicas como getchar() ou scanf("%*c") para tratar o newline.
- Não deixe para fazer o trabalho de ultima hora! Em caso de dúvidas, compareça à monitoria ou pergunte em aula.

Exemplo de árvore em memória primária:
Dada a instância com os seguintes dados inseridos pela seguinte ordem:
(RA, Nome, IRA, Status da Matrícula)
444444 – Mariazinha da Silva – 13200 - 1
555555 – Joãzinho Cardoso – 920 - 1
555551 – Pedrinho Bezerra – 15700 - 1
123321 – Luizinho Oliveira – 99999 - 0

A organização em memória deverá ser algo como:



Para salvar as árvores em disco utilize a travessia de pré ordem. Seguindo o seguinte padrão: "RA#IRA#ATIVO#NOME\n" – para algum nó não nulo e "#\n" para nós nulos. As respectivas árvores do exemplo a cima em disco ficariam:

```
ra.dat:
444444#13200#1#Mariazinha da Silva
123321#99999#0#Luizinho Oliveira
#
#
555555#920#1#Joãozinho Cardoso
555551#15700#1#Pedrinho Bezerra
#
#
#
```

nome.dat:
444444#13200#1#Mariazinha da Silva
555555#920#1#Joãozinho Cardoso
#
123321#99999#0#Luizinho Oliveira
#
#
555551#15700#1#Pedrinho Bezerra
#
#