

Trabalho 02

(individual – entregue via Run Codes)

Escreva, em linguagem C pura (sem nenhum comando próprio de C++), um programa para armazenar em um arquivo binário as seguintes informações sobre alunos:

- Nome completo: string com um máximo de 29 caracteres.
- Sigla do curso: string com um máximo de 3 caracteres.
- Número de matrícula: inteiro.

Estes dados devem ser armazenados sequencialmente dentro do arquivo binário, isto é, pela ordem de inserção. Porém, para acelerar a busca pelas informações de um aluno dentro do arquivo, implemente uma tabela hash, com linear probing, que guarde o nome do aluno (chave) e o seu índice dentro do arquivo. Assim, quando o usuário realizar a busca, sempre pelo nome do aluno, primeiro busca-se este aluno na tabela hash e, se ele estiver inserido na estrutura, obtém-se o seu índice dentro do arquivo. A seguir, é só posicionar o ponteiro de leitura na posição indicada pelo índice e recuperar o restante dos dados do aluno (curso e matrícula). Como função de espalhamento, use o método da divisão, sendo que este deve receber como parâmetro o valor inteiro que representa a string do nome (resposta da função `chaveString()` vista em aula).

Inicialmente, seu programa deve ler um valor inteiro N representando o tamanho da tabela hash e realizar a alocação dinâmica dessas N posições. A seguir, leia o nome do arquivo onde os dados serão armazenados. Por fim, leia uma opção do usuário para realizar uma das seguintes ações:

- Se a opção lida for **1**, seu programa deve ler os dados de um novo aluno e armazená-lo no arquivo e na tabela hash. Os dados são: nome completo, sigla do curso e número de matrícula (nesta ordem).
- Se a opção lida for **2**, seu programa deve ler o nome completo de um aluno e pesquisá-lo na tabela hash. Se o aluno pesquisado estiver presente na tabela, imprima seu número de matrícula, nome completo e sigla do curso, em uma única linha, separados entre si por um único espaço. Se o aluno pesquisado não estiver presente na tabela, imprima um “x” minúsculo como resposta. Pule uma linha após imprimir a resposta.
- Se a opção lida for **3**, seu programa deve ler o nome completo de um aluno e removê-lo da tabela hash, caso ele esteja presente. Caso o aluno não esteja presente na estrutura, nenhuma ação é realizada. A remoção ocorre apenas na tabela hash. Não se apaga dados de arquivo. Os bytes do arquivo que contém informações sobre alunos removidos são simplesmente sobrescritos quando outro aluno for inserido naquela mesma posição.
- Se a opção lida for **4**, seu programa deve imprimir toda a tabela hash no seguinte formato:

```
[0]: 20210002 Pedro Missioneiro CCO 1
[1]: vazio
[2]: vazio
[3]: 20210001 Ana Terra SIN 0
[4]: removido
[5]: vazio
```

Onde, entre colchetes está a posição do vetor e, a seguir, o conteúdo: a posição pode estar vazia ou ter sido removida. Se um elemento estiver presente, abra o arquivo e recupere os seus dados para impressão: imprima a matrícula, nome completo, sigla do curso e índice do registro no arquivo (separados por um único espaço).

- Se a opção lida for **9**, seu programa deve finalizar. Note que seu programa encerra apenas nesta opção. Ao terminar de processar qualquer outra opção, o programa deve retornar ao loop inicial para ler a próxima ação que o usuário deseja executar.

Relacionamento entre a tabela hash e o arquivo de dados

- Inicialmente, a tabela está vazia. Seja o primeiro aluno a ser inserido <Ana Terra, SIN, 20210001> e, suponha que a função de espalhamento o mapeou para a posição 3 da tabela:

Tabela hash: <nome, índice do registro>

[0]	[1]	[2]	[3]	[4]	[...]
vazio	vazio	vazio	<Ana Terra, 0>	vazio	vazio

Arquivo: <nome, sigla do curso, matrícula>

índice do registro	[0]			[1]
bytes do arquivo	0-29	30-33	34-37	38
conteúdo	Ana Terra	SIN	20210001	

- Ao inserir o segundo aluno <Pedro Missioneiro, CCO, 20210002>, suponha que a função de espalhamento o mapeou para a posição 0 da tabela:

Tabela hash: <nome, índice do registro>

[0]	[1]	[2]	[3]	[4]	[...]
<Pedro Missioneiro, 1>	vazio	vazio	<Ana Terra, 0>	vazio	vazio

Arquivo: <nome, sigla do curso, matrícula>

índice do registro	[0]			[1]			[2]
bytes do arquivo	0-29	30-33	34-37	38-67	68-71	72-75	76
conteúdo	Ana Terra	SIN	20210001	Pedro Missioneiro	CCO	20210002	

- Seja a remoção de <Ana Terra>:

Tabela hash: <nome, índice do registro>

[0]	[1]	[2]	[3]	[4]	[...]
<Pedro Missioneiro, 1>	vazio	vazio	removido	vazio	vazio

Arquivo: <nome, sigla do curso, matrícula>

índice do registro	[0]			[1]			[2]
bytes do arquivo	0-29	30-33	34-37	38-67	68-71	72-75	76
conteúdo	Ana Terra	SIN	20210001	Pedro Missioneiro	CCO	20210002	

- Ao inserir o novo aluno <Jose Borges, ADM, 20210003>, suponha que a função de espalhamento o mapeou para a posição 3 da tabela. Como esta posição indica que o elemento foi removido, o novo aluno é inserido nesta posição. Além disso, o elemento que foi removido (Ana Terra) está gravado como o registro 0 do arquivo. Neste caso, o registro 0 do arquivo é sobrescrito com os dados do novo aluno.
- Ou seja, ao inserir numa posição removida do hash, você não deve aumentar o arquivo com 1 registro a mais, e sim sobrescrever no arquivo o registro que foi removido.*

Tabela hash: <nome, índice do registro>

[0]	[1]	[2]	[3]	[4]	[...]
<Pedro Missioneiro, 1>	vazio	vazio	<Jose Borges, 0>	vazio	vazio

Arquivo: <nome, sigla do curso, matrícula>

índice do registro	[0]			[1]			[2]
bytes do arquivo	0-29	30-33	34-37	38-67	68-71	72-75	76
conteúdo	Jose Borges	ADM	20210003	Pedro Missioneiro	CCO	20210002	