



INSTITUTO FEDERAL
CATARINENSE
Câmpus Videira

Turma: Ciência da Computação 3º Semestre

Aluno: Felipe Veroneze

Disciplina: Estrutura de Dados I

Professor: Manassés Ribeiro

Relatório Trabalho Final

O código em C manipula uma tabela hash com listas encadeadas para armazenar nomes. O programa oferece várias funcionalidades, como adicionar nomes, pesquisar nomes, remover nomes, imprimir os nomes por lista, imprimir todos os nomes, contar o total de nomes em uma lista, contar o total de nomes na tabela e ordenar os nomes usando o algoritmo de ordenação quicksort.

O programa consiste em duas listas encadeadas que servem para manipular as chaves da tabela e a outra para armazenar os nomes.

```
✓ typedef struct sNodo {  
    struct sNodo *next;  
    struct sNodo *prev;  
    char dado[30];  
} Nodo;  
  
✓ typedef struct sLista {  
    struct sNodo *head;  
    struct sNodo *tail;  
    struct sLista *next;  
    struct sLista *prev;  
    int chave;  
    int size;  
} Lista;  
  
✓ typedef struct sTabela {  
    struct sLista *head;  
    struct sLista *tail;  
    int chave;  
    int size;  
} Tabela;
```

Os nomes são inseridos através da função `insereNome()`, nela é feita a função Hash, é buscada a lista com o resultado desta função, e inserido na tabela como uma função de inserir de listas encadeadas duplas. Vale ressaltar que o código pega um nome, passa pela função hash e baseado nisso aponta para qual chave o nome ficará inserido.

```
int funcaoHash(char *dado) {  
    int i, chave = 0;  
    for (i = 0; i < strlen(dado); i++) {  
        chave += dado[i] * 7;  
    }  
    return (chave % M);  
}
```

```
void insereNome(Tabela *tabela, char *dado) {  
    int chave = funcaoHash(dado);  
    Lista *lista = buscaLista(tabela, chave);  
    insereTabela(tabela, lista, dado);  
}
```

Ao executar o programas você pode escolher entre as diversas opções:

```
1 = Adicionar nome  
2 = Pesquisar nome  
3 = Remover nome  
4 = Imprime nomes por lista  
5 = Imprimir todos os nomes  
6 = Total de nomes em uma lista  
7 = Total de nomes  
8 = Quick sort dos nomes
```

Você pode adicionar um nome, buscá-lo e removê-lo. A opção 4 imprime os nomes que estiverem na lista que você irá informar, como são 53 chaves, o número será entre 0 e 52.

```
LIONICE
GEROZINO
VALMICELIA
NARRANA
THAMYRES
MARVIO
JORVAL
GERVISON
ADEVANILDO
AGUIBERTO
OSMERINDA
MATTHEWS
EROMIR
WUILLA
KEZZIA
RTDALVA
AVALDIR
JOSSAN
TAIUANA
TITUS
ELIAUDO
RAMILY
LAURISON
DEIDISSON
ELIESEL
DIERLEN
HYAM
ENICE
DENIELTON
CLESTIANO
NATHANI
HUIGOR
JABER
CALEO
NILL
ASSENILDO
ARGIA
THAISIA
AARAO
NEICE
NATHALLYE
JESSIELEN
```

A lista gerada não estará ordenada, pois para fazer a ordenação é preciso usar a oitava opção, a opção 5 imprime todos os nomes, a 6 o total de nomes que há em uma chave, e a 7 o total de nomes existentes na tabela inteira.

```
Lista: 2
```

```
Quantidade de nomes na lista 2: 1880
```

Resultado após o QuickSort:

```
WILKS  
WILLAMES  
WILLIANN  
WILLTON  
WILQUER  
WIRES  
WISLLEY  
WITINEY  
WLLITON  
WOLACI  
WOLGRAND  
WOODSON  
WOSHINTOM  
WUARLEY  
WUELITO  
WUIDSON  
WUILI  
WUILIANE  
WYLLAMS  
WYLLYS  
WYLMA  
YANDRA  
YANICK  
YIGOR  
YURIA  
ZAHIRA  
ZALENE  
ZALINA  
ZANATA  
ZEDENI  
ZELANE  
ZELENA  
ZENEDI  
ZENIDE  
ZENILDES  
ZENIT  
ZENON  
ZILANA  
ZILVANIA  
ZILZA  
ZIRLAINE  
ZULENIR
```

O Quick Sort utilizado foi o método de Lomuto, apesar do que foi visto em sala, de que o de Hoare pode nos dar resultados mais rapidamente, o código com Lomuto funcionou de uma forma ágil.

Como já citado, as inserções, remoções e buscas, são baseadas na estrutura das listas duplamente encadeadas, assim ao ter uma lista de chaves cada chave terá uma lista de nomes, facilitando assim o tratamento de colisão. Esse tratamento só é possível ser feito graças a lista encadeada.

Trabalhar com este volume de dados com o uso da tabela hash auxilia principalmente na busca, pois com esses dados organizados com um padrão, basta calcular em qual chave esse dado está e não será necessário passar por todos eles.

Por fim, abaixo segue um gráfico contendo todas as chaves e sua quantidade de nomes em cada uma. Algo a ser considerado é a uniformidade na quantidade de nomes nas chaves, os valores não se distanciam tanto um do outro.

