



Algoritmo e Estrutura de Dados II

Profª. Vanessa Souza

Assunto: Tabela Hash

Questão 1:

A Tabela Hash pode ser implementada como um *array* de ponteiros para listas. Neste caso, todas as colisões são tratadas inserindo o novo elemento na lista daquela posição da Hash.

Nesta atividade vocês deverão fazer uso da TAD Lista simplesmente encadeada para implementar uma tabela Hash que guarda palavras. A função de hash é calculada da seguinte forma:

1. Soma o código ASCII de todas as letras da palavra, que deve ter tamanho máximo de 30 caracteres.
2. Retorna o resto da divisão (MOD) da soma pelo tamanho da hash.

Exemplo:

- Palavra = "vanessa"
- Soma do código ASCII => 753
- Uma tabela de 15 posições resultaria na posição 3 => (753 MOD 15)

Dessa forma, a palavra vanessa seria inserida na lista da posição 3 da tabela hash.

Não é necessário manter uma lista ordenada. Pode inserir sempre no início da lista.

Nessa atividade, portanto, haverá duas TADs. A de lista e a de hash. O .h de ambas, a main e um arquivo de entrada para teste estão disponíveis. Nas figuras abaixo a struct hash, no e lista. O arquivo de entrada apresenta na primeira linha o tamanho da hash.

```
struct hash
{
    lista **tabela;
    int tamanho;
};
```

```
struct no{
    char palavra[30];
    struct no *prox;
};
```

```
struct lista{
    struct no *inicio;
    int tam;
};
```

Abaixo a saída que deve ser produzida utilizando a main e o arquivo de entrada disponibilizados na atividade.

Saída Esperada do Caso de Teste

```
Posicao 0 contem 2 palavras
Posicao 1 contem 4 palavras
Posicao 2 contem 2 palavras
Posicao 3 contem 3 palavras
Posicao 4 contem 2 palavras
Posicao 5 contem 5 palavras
Posicao 6 contem 2 palavras
Posicao 7 contem 0 palavras
Posicao 8 contem 3 palavras
Posicao 9 contem 4 palavras
Posicao 10 contem 1 palavras
Posicao 11 contem 1 palavras
Posicao 12 contem 3 palavras
Posicao 13 contem 2 palavras
Posicao 14 contem 2 palavras

Elemento encontrado na hash. Posicao 9.

Elemento não encontrado na hash
ABA
ABACATE
ABAFA
INTERINO

INTERLINGUISTICO
ABALAVEL

Palavra não encontrada na hash
```

ATENÇÃO: O código *hash.c* deve ser entregue no **run.codes**. É de extrema importância que a saída seja formatada da forma ilustrada acima para que o **run.codes** corrija corretamente seu código. O código *lista.c* deve ser entregue no SIGAA para correção, uma vez que no run.codes não é possível submeter mais de um arquivo por atividade. A não entrega de qualquer um dos arquivos invalida a atividade.

A atividade pode ser feita em dupla. Apenas um integrante faz a entrega. Os integrantes da dupla devem estar informados no cabeçalho dos arquivos.