

# Projeto prático 3

SCC 0606 – Estrutura de Dados II  
Prof. Maria Cristina Ferreira de Oliveira  
20 de maio de 2024

## 1. Descrição

Guilherme adora ler textos, tanto da internet como de livros, sempre em busca de aprender palavras novas, porém são tantos textos e tantas palavras que ele já perdeu muito tempo tentando lembrar todas! Você consegue escrever um programa para ajudá-lo a contabilizar as palavras a medida em que for aprendendo?

Ele vai informar os textos, cujas palavras deverão ser armazenadas em um dicionário, que consiste basicamente em uma tabela *hash* na qual cada palavra é armazenada em uma posição segundo uma função de espalhamento, juntamente com a contagem de número de ocorrências da palavra. Uma mesma palavra não deve ser inserida múltiplas vezes no dicionário, deve-se apenas incrementar seu contador a cada nova ocorrência. Serão solicitadas operações de inserção, remoção e busca no dicionário. Lembre-se que colisões podem ocorrer, i.e., palavras distintas podem ser mapeadas no mesmo endereço. Utilize o **hash estático fechado** e a estratégia de **sondagem quadrática** para tratar as colisões, considerando as constantes  $c_1$  e  $c_2$ , que serão lidas pelo programa.

Considerando que cada palavra é um *string*, e  $S$  é o tamanho da tabela, utilize a seguinte função *hash* para mapear uma palavra a um determinado endereço da tabela:

```
hash(palavra,S):  
    inteiro v = 0;  
    para cada caractere em palavra faça:  
        v += v*3 + ascii(caractere);  
        v = v % S;  
    fim para  
    retorna v;
```

## 2. Entrada e Saída

Primeiramente, seu programa receberá na entrada  $S$ ,  $C_1$  e  $C_2$ , que informam, respectivamente, o tamanho da tabela *hash*, e os parâmetros da sondagem quadrática. Em seguida, uma sequência de inteiros  $T$ , em que  $T$  são valores entre 0 e 5. Se  $T = 0$  o programa é encerrado, se  $T \neq 0$  o processamento correspondente ao valor de  $T$  (como descrito a seguir) é realizado e um novo valor  $T$  é lido na sequência.

T=1: Lê em seguida uma linha de texto até atingir o caractere ascii '\n', adicionando cada palavra no dicionário.

- É garantido nos casos de teste que todas as palavras serão em letras minúsculas, sem caracteres fora da tabela ascii, e separadas por espaços (acentos, pontuações, maiúsculas e caracteres especiais não ocorrem).
- Não há nada a ser exibido na saída deste comando, ele somente afeta a configuração do dicionário.
- Se algum texto já foi lido anteriormente, mantenha o mesmo dicionário, o qual deve ser atualizado para incluir as palavras que ocorrem no novo texto.
- Essa opção recebe no máximo 100.000 caracteres.
- O tamanho máximo de uma palavra é 30 caracteres.
- Exemplo:

Entrada
1 o que temos que ter sempre em mente e que a continua expansao de nossa atividade possibilita uma melhor visao global do sistema de participacao geral

T=2: Exibe o número de palavras diferentes que constam no dicionário até agora, na próxima linha exibe a palavra mais frequente do dicionário e quantas vezes ela aparece.

- Caso haja mais de uma palavra, mostre somente a que vem primeiro em ordem alfabética (ou ordem lexicográfica).
- Se não houver nenhuma palavra, mostre apenas “nenhuma palavra encontrada”.
- Exemplo:

Entrada	Saída
2	foram encontradas 104 palavras diferentes palavra mais frequente = gatinho, encontrada 20 vezes

(pule uma linha entre as duas frases)

T=3: Lê um inteiro n, as próximas n linhas contém uma palavra cada, para cada palavra que esteja no dicionário a saída exibe <palavra> encontrada e em seguida o número de ocorrências, ou caso contrário exibe “<palavra> nao encontrada”.

- Exemplo:

Entrada	Saída
3 5	astronauta encontrada 1 lousa nao encontrada

astronauta lousa computadores turing cachorro	computadores encontrada 10 turing encontrada 3 cachorro encontrada 2
---	--

T=4: Lê um inteiro n, as próximas n linhas contém uma palavra cada, a qual deve ser removida do dicionário caso seja encontrada, do contrário nada precisa ser feito. A saída exibe <palavra> removida caso a palavra tenha sido encontrada, ou do contrário exibe <palavra> nao encontrada.

- Exemplo:

Entrada	Saída
4 5 mamao banana abacaxi banana uva	mamao removida banana removida abacaxi nao encontrada banana nao encontrada uva removida

T=5: Exibe todas as palavras contidas no dicionário e suas respectivas posições na tabela *hash*.

- Exemplo (considerando o texto do exemplo T=1, para uma tabela de tamanho S=89, C1=2, C2=2):

Entrada	Saída
5	imprimindo tabela hash participacao 1 uma 2 sistema 6 a 8 e 12 nossa 14 temos 15 visao 18 o 22 melhor 23 mente 25 continua 27 atividade 29 possibilita 37 sempre 39 de 56 ter 60 que 63

	global 66 em 68 do 70 expansao 71 geral 86 fim da tabela hash
--	--

T=0: Encerra o programa.

### 3. Conteúdo e data de entrega

O trabalho é individual e a data limite de submissão é **12/06/2024** (trabalhos em atraso não serão aceitos), no sistema RunCodes (<https://runcodes.icmc.usp.br/>). O formato da entrega deve ser um arquivo com o código fonte desenvolvido.

- O código deverá estar adequadamente comentado/documentado;
- O trabalho pode ser desenvolvido utilizando C ou Python, a escolha;
- Não é permitido utilizar bibliotecas que já disponibilizam os algoritmos de *hashing* de forma parcial ou total;
- Todo o código deve ser desenvolvido pelo próprio aluno. Os códigos serão submetidos a um detector de plágio, e a utilização de código criado por terceiros será tratada como plágio (nota zero).

Dicas:

(i) quem vai programar em C pode utilizar funções da biblioteca string.h (como strcpy, strcat ou strcmp) para facilitar as operações com strings, inclusive a função strtok para facilitar a separação das palavras do texto;

(ii) caso seu programa esteja gerando alguma saída errada, mas está difícil identificar a diferença entre a saída esperada e a resultante, este site para comparação de textos pode ser útil: <https://www.diffchecker.com/text-compare/>

### 4. Critérios de avaliação

- 9,0 pontos pela implementação dos algoritmos solicitados, de forma correta (avaliado segundo o resultado do RunCodes, e por inspeção do código).
- 1,0 ponto por código legível, bem estruturado e bem comentado.

## 5. Exemplos de entrada e saída

1)

Entrada	Saída
503 2 2 o rato roeu a roupa do rei de roma 2 3 2 rato hamster 4 1 rato 2 0	foram encontradas 9 palavras diferentes palavra mais frequente = a, encontrada 1 vezes rato encontrada 1 hamster nao encontrada rato removida foram encontradas 8 palavras diferentes palavra mais frequente = a, encontrada 1 vezes

2)

Entrada	Saída
1663 2 2 hash functions and their associated hash tables are used in data storage and retrieval applications to access data in a small and nearly constant time per retrieval They require an amount of storage space only fractionally greater than the total space required for the data or records themselves hashing is a computationally and storage space efficient form of data access that avoids the non constant access time of ordered and unordered lists and structured trees and the often exponential storage requirements of direct access of state spaces of large or variable length keys 2 3 5 functions abacaxi and hash data 4 5 batatadoce and computationally computationally 2 0	foram encontradas 61 palavras diferentes palavra mais frequente = and, encontrada 7 vezes functions encontrada 1 abacaxi nao encontrada and encontrada 7 hash encontrada 2 data encontrada 4 batatadoce nao encontrada and removida Hash removida computationally removida computationally nao encontrada foram encontradas 58 palavras diferentes palavra mais frequente = of, encontrada 6 vezes

3)

Entrada	Saída
---------	-------

109 2 2 the quick brown fox jumps over the lazy dog 1 minha casa e de madeira 1 subi no onibus 1 cachorro cachorro cachorro cachorro cachorro cachorro cachorro cachorro cachorro cachorro cachorro 3 1 cachorro 4 3 matematica geometria cachorro 2 0	cachorro encontrada 11 matematica nao encontrada geometria nao encontrada cachorro removida foram encontradas 16 palavras diferentes palavra mais frequente = the, encontrada 2 vezes
---	--

4)

Entrada	Saída
4909 2 2 alan mathison turing was an english mathematician computer scientist logician cryptanalyst philosopher and 2 1 theoretical biologist turing was highly influential in the development of theoretical computer science providing 2 1 a formalisation of the concepts of algorithm and computation with the turing machine which can be considered 2 1 a model of a general-purpose computer he is widely considered to be 2 1 the father of theoretical computer science and artificial intelligence 2 1 the father of theoretical computer science and artificial intelligence 2 0	foram encontradas 13 palavras diferentes palavra mais frequente = alan, encontrada 1 vezes foram encontradas 23 palavras diferentes palavra mais frequente = computer, encontrada 2 vezes foram encontradas 34 palavras diferentes palavra mais frequente = of, encontrada 3 vezes foram encontradas 41 palavras diferentes palavra mais frequente = of, encontrada 4 vezes foram encontradas 44 palavras diferentes palavra mais frequente = of, encontrada 5 vezes foram encontradas 44 palavras diferentes palavra mais frequente = of, encontrada 6 vezes

5)

Entrada	Saída
<p>91 2 2 1 alan mathison turing was an english mathematician computer scientist logician cryptanalyst philosopher and theoretical biologist turing was highly influential in the development of theoretical computer science providing a formalisation of the concepts of algorithm and computation with the turing machine which can be considered a model of a general purpose computer he is widely considered to be the father of theoretical computer science and artificial intelligence 5 0</p>	<p>imprimindo tabela hash of 0 concepts 1 turing 6 scientist 7 mathematician 10 the 11 is 13 influential 17 a 18 father 20 artificial 24 computation 25 english 26 to 29 development 35 with 36 be 38 intelligence 40 was 41 an 43 science 46 highly 47 computer 52 formalisation 56 widely 57 general 58 alan 62 algorithm 64 considered 65 philosopher 66 machine 70 he 74 in 75 biologist 78 which 79 can 80 providing 81 mathison 82 model 83 cryptanalyst 84 logician 86 purpose 87 theoretical 88 and 90 fim da tabela hash</p>

## 6. Passo a passo de execução de um exemplo

Segue abaixo o passo a passo de execução de um novo exemplo:

Entrada	Saída
15 2 2 1 can he drive 1 can she fly 1 if possible 4 2 car can 1 if he can 5 2 0	car nao encontrada can removida imprimindo tabela hash if 1 she 2 drive 4 he 7 fly 10 can 12 possible 14 fim da tabela hash foram encontradas 7 palavras diferentes palavra mais frequente = he, encontrada 2 vezes

Atividade a ser executada	Lógica	Tabela Hash após a execução																											
15 2 2	Inicialização da hash com 15 posições de memória vazias e constantes c1 e c2 da sondagem quadrática	<table> <tr> <th>Índice</th><th>Palavra</th><th>Contagem</th></tr> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td></td><td>0</td></tr> <tr><td>2</td><td></td><td>0</td></tr> <tr><td>3</td><td></td><td>0</td></tr> <tr><td>4</td><td></td><td>0</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td></td><td>0</td></tr> </table>	Índice	Palavra	Contagem	0		0	1		0	2		0	3		0	4		0	5		0	6		0	7		0
Índice	Palavra	Contagem																											
0		0																											
1		0																											
2		0																											
3		0																											
4		0																											
5		0																											
6		0																											
7		0																											



		<table> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td></td><td>0</td></tr> <tr><td>11</td><td></td><td>0</td></tr> <tr><td>12</td><td></td><td>0</td></tr> <tr><td>13</td><td></td><td>0</td></tr> <tr><td>14</td><td></td><td>0</td></tr> </table>	8		0	9		0	10		0	11		0	12		0	13		0	14		0																											
8		0																																																
9		0																																																
10		0																																																
11		0																																																
12		0																																																
13		0																																																
14		0																																																
1 can he drive	<p>Inserção das palavras “can”, “he” e “drive”.</p> <p>Hash(can) = 12. Posição 12 nunca foi preenchida, logo, insere “can” no índice 12.</p> <p>Hash(he) = 7. Posição 7 nunca foi preenchida, logo, insere “he” no índice 7.</p> <p>Hash(drive) = 4. Posição 4 nunca foi preenchida, logo, insere “drive” no índice 4.</p>	<table> <tr><th>Índice</th><th>Palavra</th><th>Contagem</th></tr> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td></td><td>0</td></tr> <tr><td>2</td><td></td><td>0</td></tr> <tr><td>3</td><td></td><td>0</td></tr> <tr><td>4</td><td>drive</td><td>1</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td>he</td><td>1</td></tr> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td></td><td>0</td></tr> <tr><td>11</td><td></td><td>0</td></tr> <tr><td>12</td><td>can</td><td>1</td></tr> <tr><td>13</td><td></td><td>0</td></tr> <tr><td>14</td><td></td><td>0</td></tr> </table>	Índice	Palavra	Contagem	0		0	1		0	2		0	3		0	4	drive	1	5		0	6		0	7	he	1	8		0	9		0	10		0	11		0	12	can	1	13		0	14		0
Índice	Palavra	Contagem																																																
0		0																																																
1		0																																																
2		0																																																
3		0																																																
4	drive	1																																																
5		0																																																
6		0																																																
7	he	1																																																
8		0																																																
9		0																																																
10		0																																																
11		0																																																
12	can	1																																																
13		0																																																
14		0																																																
1 can she fly	<p>Inserção das palavras “can”, “she” e “fly”.</p> <p>Hash(can) = 12. Posição 12 já foi preenchida, com a palavra “can”, logo, adiciona 1 ao contador.</p> <p>Hash(she) = 2. Posição 2 nunca foi preenchida, logo, insere “she” no índice 2.</p> <p>Hash(fly) = 10. Posição 10 nunca foi</p>	<table> <tr><th>Índice</th><th>Palavra</th><th>Contagem</th></tr> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td></td><td>0</td></tr> <tr><td>2</td><td>she</td><td>1</td></tr> <tr><td>3</td><td></td><td>0</td></tr> </table>	Índice	Palavra	Contagem	0		0	1		0	2	she	1	3		0																																	
Índice	Palavra	Contagem																																																
0		0																																																
1		0																																																
2	she	1																																																
3		0																																																

	preenchida, logo, insere "fly" no índice 10.	<table> <tr><td>4</td><td>drive</td><td>1</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td>he</td><td>1</td></tr> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td>fly</td><td>1</td></tr> <tr><td>11</td><td></td><td>0</td></tr> <tr><td>12</td><td>can</td><td>2</td></tr> <tr><td>13</td><td></td><td>0</td></tr> <tr><td>14</td><td></td><td>0</td></tr> </table>	4	drive	1	5		0	6		0	7	he	1	8		0	9		0	10	fly	1	11		0	12	can	2	13		0	14		0															
4	drive	1																																																
5		0																																																
6		0																																																
7	he	1																																																
8		0																																																
9		0																																																
10	fly	1																																																
11		0																																																
12	can	2																																																
13		0																																																
14		0																																																
1 if possible	<p>Inserção das palavras "if" e "possible".</p> <p>Hash(if) = 12. Posição 12 já foi preenchida, com a palavra "can", logo, precisamos rodar o código da colisão. Colisão(if) = 1. Posição 1 nunca foi preenchida, logo, insere "if" no índice 1.</p> <p>Hash(possible) = 10. Posição 10 já foi preenchida, com a palavra "fly", logo, precisamos rodar o código da colisão. Colisão(possible) = 14. Posição 14 nunca foi preenchida, logo, insere "possible" no índice 14.</p>	<table> <tr><th>Índice</th><th>Palavra</th><th>Contagem</th></tr> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td>if</td><td>1</td></tr> <tr><td>2</td><td>she</td><td>1</td></tr> <tr><td>3</td><td></td><td>0</td></tr> <tr><td>4</td><td>drive</td><td>1</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td>he</td><td>1</td></tr> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td>fly</td><td>1</td></tr> <tr><td>11</td><td></td><td>0</td></tr> <tr><td>12</td><td>can</td><td>2</td></tr> <tr><td>13</td><td></td><td>0</td></tr> <tr><td>14</td><td>possible</td><td>1</td></tr> </table>	Índice	Palavra	Contagem	0		0	1	if	1	2	she	1	3		0	4	drive	1	5		0	6		0	7	he	1	8		0	9		0	10	fly	1	11		0	12	can	2	13		0	14	possible	1
Índice	Palavra	Contagem																																																
0		0																																																
1	if	1																																																
2	she	1																																																
3		0																																																
4	drive	1																																																
5		0																																																
6		0																																																
7	he	1																																																
8		0																																																
9		0																																																
10	fly	1																																																
11		0																																																
12	can	2																																																
13		0																																																
14	possible	1																																																

<p>4 2 car can</p>	<p>Remove as palavras “car” e “can”.</p> <p>Hash(car) = 5. Como a posição 5 nunca foi preenchida, nunca foi inserida a palavra “car”. Assim sendo, imprime: “car não encontrada”</p> <p>Hash(can) = 12. A posição 12 possui a palavra “can”, logo, removemos a palavra da posição 12, deixando claro que tal posição já foi ocupada, usando algum indicador (neste caso “###”), e imprime: “can removida”</p>	<table border="1"> <thead> <tr> <th>Índice</th><th>Palavra</th><th>Contagem</th></tr> </thead> <tbody> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td>if</td><td>1</td></tr> <tr><td>2</td><td>she</td><td>1</td></tr> <tr><td>3</td><td></td><td>0</td></tr> <tr><td>4</td><td>drive</td><td>1</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td>he</td><td>1</td></tr> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td>fly</td><td>1</td></tr> <tr><td>11</td><td></td><td>0</td></tr> <tr><td>12</td><td>###</td><td>0</td></tr> <tr><td>13</td><td></td><td>0</td></tr> <tr><td>14</td><td>possible</td><td>1</td></tr> </tbody> </table>	Índice	Palavra	Contagem	0		0	1	if	1	2	she	1	3		0	4	drive	1	5		0	6		0	7	he	1	8		0	9		0	10	fly	1	11		0	12	###	0	13		0	14	possible	1
Índice	Palavra	Contagem																																																
0		0																																																
1	if	1																																																
2	she	1																																																
3		0																																																
4	drive	1																																																
5		0																																																
6		0																																																
7	he	1																																																
8		0																																																
9		0																																																
10	fly	1																																																
11		0																																																
12	###	0																																																
13		0																																																
14	possible	1																																																
<p>1 if he can</p>	<p>Inserção das palavras “if”, “he” e “can”.</p> <p>Hash(if) = 12. Posição 12 já foi preenchida, anteriormente, pois contém “###”, logo, precisamos verificar se existe outra posição que já possui a palavra “if”, antes de inserir esta na posição 12. Precisamos então rodar o código da colisão até achar a palavra “if” ou encontrar uma posição nunca antes ocupada por nenhuma palavra.. Colisão(if) = 1. Posição 1 possui a palavra “if”, logo, incrementa o valor da contagem da posição 1.</p> <p>Hash(he) = 7. Posição 7 possui a palavra “he”, logo, incrementa o valor da contagem da posição 7.</p> <p>Hash(can) = 12. Posição 12 já foi preenchida, anteriormente, pois contém “###”, logo, precisamos verificar se existe outra posição que já possui a palavra “can”, antes de inserir esta na posição 12. Precisamos então rodar o código da colisão até achar a palavra “can” ou encontrar uma posição nunca antes ocupada por nenhuma palavra.. Colisão(can) = 1. Posição 1 possui a</p>	<table border="1"> <thead> <tr> <th>Índice</th><th>Palavra</th><th>Contagem</th></tr> </thead> <tbody> <tr><td>0</td><td></td><td>0</td></tr> <tr><td>1</td><td>if</td><td>2</td></tr> <tr><td>2</td><td>she</td><td>1</td></tr> <tr><td>3</td><td></td><td>0</td></tr> <tr><td>4</td><td>drive</td><td>1</td></tr> <tr><td>5</td><td></td><td>0</td></tr> <tr><td>6</td><td></td><td>0</td></tr> <tr><td>7</td><td>he</td><td>2</td></tr> <tr><td>8</td><td></td><td>0</td></tr> <tr><td>9</td><td></td><td>0</td></tr> <tr><td>10</td><td>fly</td><td>1</td></tr> <tr><td>11</td><td></td><td>0</td></tr> </tbody> </table>	Índice	Palavra	Contagem	0		0	1	if	2	2	she	1	3		0	4	drive	1	5		0	6		0	7	he	2	8		0	9		0	10	fly	1	11		0									
Índice	Palavra	Contagem																																																
0		0																																																
1	if	2																																																
2	she	1																																																
3		0																																																
4	drive	1																																																
5		0																																																
6		0																																																
7	he	2																																																
8		0																																																
9		0																																																
10	fly	1																																																
11		0																																																

	<p>palavra “if”, logo, precisamos aplicar novamente a função de colisão. Colisão(can) = 9. Posição 9 nunca foi preenchida, portanto, a tabela não possui a palavra “can”, então preenchamos esta no seu primeiro lugar vago, ou seja, na posição 12.</p>	<table> <tr> <td>12</td><td>can</td><td>1</td></tr> <tr> <td>13</td><td></td><td>0</td></tr> <tr> <td>14</td><td>possible</td><td>1</td></tr> </table>	12	can	1	13		0	14	possible	1
12	can	1									
13		0									
14	possible	1									
<p>5 2 0</p>	<p>5 é o comando que imprime a posição de todos os valores do hash:</p> <p>“imprimindo tabela hash if 1 she 2 drive 4 he 7 fly 10 can 12 possible 14 fim da tabela hash”</p> <p>2 é o comando que imprime a quantidade de palavras na hash e qual delas possui maior frequência. Repare que tanto “if” quanto “he” aparecem duas vezes, mas imprimimos apenas a “menor” palavra lexicograficamente. Logo a saída fica:</p> <p>“foram encontradas 7 palavras diferentes palavra mais frequente = he, encontrada 2 vezes”</p> <p>Comando 0 encerra o programa.</p>	<p>Programa encerrado, tabela desalocada da memória</p>									