

**sqtpm**

[jhdapr]

[voltar](#)**Trabalho:** 25-hashing

Data de abertura: 28/10/2021 18:00:00

Data limite para envio: 10/11/2021 23:59:59 (aberto)

Número máximo de envios: 31

Casos-de-teste abertos: [casos-de-teste.tgz](#)

Arquivos a enviar: entre 1 e 3

**Enviar:**

Linguagem:

Arquivos:

Nenhum arquivo selecionado

## Hashing de cadeias

Este trabalho consiste em implementar uma tabela de hashing para armazenar cadeias de caracteres. Cada cadeia que for armazenada deve receber um timestamp (que é um número natural único e seqüencial, a partir de 0 e sem reuso) que indica o momento que em a cadeia foi inserida na tabela.

As colisões devem ser resolvidas usando sondagem com incremento por hashing dupla. Escolha uma função de hashing para a sua tabela.

Escolha também uma função de transformação de cadeias de caracteres em naturais. Não importa a forma como você faça para este trabalho, mas algumas funções são muito piores que outras em termos de espalhamento. Um exemplo de uma função que funciona bem na prática é a função djb2:

```
unsigned long djb2(unsigned char *str) {
    unsigned long hash = 5381;
    int c;

    while ((c = *str++))
        hash = ((hash << 5) + hash) ^ c; // hash * 33 XOR c

    return hash;
}
```

A djb2 faz um mapeamento bem espalhado de strings em unsigned long. Observe que o retorno da djb2 pode ser um número muito maior que o tamanho da tabela de hashing.

## Entrada

Cada linha da entrada para o programa é um comando. Os comandos podem ser

- 'c n'

Seu programa deve criar uma tabela vazia para armazenar n dados. O tamanho da tabela deve ser o menor número primo tal que a tabela não fique mais que 80% ocupada quando tiver n dados. Se já houver

# sqtpm

[jhdapr]

## voltar

uma tabela, ela deve ser descartada e dar lugar à nova tabela vazia e o timestamp deve ser reiniciado com zero.

- 'i cadeia'

Seu programa deve inserir a cadeia na tabela com o timestamp corrente. Cada cadeia deve ser inserida uma única vez na tabela. Se houver um comando de inserção para uma cadeia que já está na tabela, o timestamp deve ser atualizado para o timestamp corrente.

Depois de um comando i o timestamp deve ser incrementado de uma unidade.

- 'r cadeia'

Seu programa deve remover a cadeia da tabela.

- 'b cadeia'

Se a cadeia estiver na tabela, seu programa deve imprimir o timestamp dela, senão deve imprimir -1.

- 'f'

Seu programa deve terminar. Qualquer memória alocada dinamicamente deve ser liberada antes do término.

Nos comandos, o caractere é separado da cadeia por um único espaço.

As cadeias são formadas por até 250 caracteres do alfabeto {a,b,...,z,A,B,...,Z,\_,-,0,1,...,9, }. Observe que uma cadeia pode começar ou terminar com um ou mais espaços e nesses casos os espaços fazem parte da cadeia.

## Saída

A saída tem a forma exemplificada abaixo.

## Exemplos

Entrada
c 32
i it doesnt matter
i the way that you take it
i it doesnt matter
i the way that you make it
b it doesnt matter
b it doesnt matter
b the way that you make it
r the way that you make it
b the way that you make it
r the way that you make it
f

Saída
TS de [it doesnt matter]: -1
TS de [it doesnt matter]: 2
TS de [the way that you make it]: 3
TS de [the way that you make it]: -1

Entrada
---------

Saída
-------

**sqtpm**

[jhdapr]

voltar

c 16

i oi

i oioi

i OiOi

i oi

i oi

b oi

b oi

b oioi

f

TS de [oi]: 4

TS de [ oi]: 3

TS de [ oioi]: -1

## Requisitos adicionais

- Não pode haver qualquer variável global. Uma variável é global se estiver declarada fora de alguma função (variáveis declaradas dentro da main não são globais, são locais à função main).

## Sobre organização do código e comentários

- Faça um programa organizado, bem indentado e que seja fácil de ler.
  - Vai ser mais fácil organizar seu programa se cada operação for realizada por uma função e cada função no programa implementar apenas uma funcionalidade.
  - Adicione comentários que vão ser úteis para entender o programa se você for relê-lo daqui a alguns anos: comentar cada linha vai ser redundante; documentar blocos de código e a estratégia usada na solução vai ser muito útil.
  - Lembre-se do princípio KISS!
-