

## Trabalho Prático

### Descrição do problema

Uma empresa de software tem como principal produto uma ferramenta de análise de textos na Web. Para isso, ela precisa de um relatório estatístico sobre os textos de entrada. Para criar esse relatório estatístico, você foi contratado. O relatório deve informar a quantidade de ocorrências das palavras mais frequentes, ordenadas pela frequência em que apareceram, além das linhas em que as palavras aparecem.

O trabalho consiste em fazer um programa que conte quantas vezes a mesma palavra aparece num texto de entrada, determinando assim a sua frequência. Uma palavra é definida como uma sequência consecutiva de letras do alfabeto, maiúsculas e/ou minúsculas. As palavras com uma única letra devem ser ignoradas, bem como espaços em brancos, sinais de pontuação, caracteres especiais, etc. Você pode assumir que as palavras não terão acentuação. Além disso, o programa deve ser case insensitive. Por exemplo, são consideradas iguais as palavras "insensitive", "InSensitive" ou "INSENSITIVE".

Para a estrutura de pesquisa e inserção das palavras deverá ser usada uma hash table. A função hash deve receber a palavra a ser inserida (código ASCII) e calcular o valor de seu hash. A função hash deve ser rápida e gerar o mínimo de colisões possível (crie diversas funções e use a melhor). Idealmente, a ocupação da tabela deve ser uniformemente distribuída. As colisões devem ser tratadas por endereçamento aberto.

### Entrada

O programa será testado da seguinte maneira: `./prog input.txt pesquisa.txt` - onde `input.txt` e `pesquisa.txt` são arquivos que contém o texto de entrada e as palavras a serem pesquisadas depois. O arquivo de entrada contém no máximo 256 palavras diferentes, cada palavra não terá mais de 20 letras, e cada linha está limitada a 80 caracteres. A entrada termina com EOF (end-of-file). O arquivo de pesquisa contém a quantidade de palavras a serem pesquisadas na primeira linha e a lista de palavras a serem pesquisadas, uma por linha.

Anexo à tarefa no Portal Didático, o arquivo `argc_e_argv.c` traz um exemplo de como rodar o programa passando os parâmetros via linha de comando.

### Saída

Para cada entrada seu programa deve produzir um conjunto de linhas na saída. Cada linha deve conter um número indicando a frequência, um espaço em branco e a palavra (todas as letras devem ser minúsculas), seguido de outro espaço e das linhas onde a palavra aparece, separadas também por espaço. Seu programa deve imprimir, na saída padrão, as palavras pesquisadas na ordem do arquivo de pesquisa, conforme o exemplo deste PDF.

## Exemplo

Entrada (input.txt)

```
Pedra, papel, tesoura, lagarto, Spock.  
Eh muito simples! Olhe - tesoura corta papel, papel cobre pedra, pedra esmaga lagarto,  
lagarto envenena Spock, Spock esmaga tesoura, tesoura decapita lagarto, lagarto come papel,  
papel refuta Spock, Spock vaporiza pedra e como sempre, pedra quebra tesoura...  
Sheldon Cooper
```

Entrada (pesquisa.txt)

```
6  
lagarto  
papel  
pedra  
spock  
tesoura  
esmaga
```

Saída no terminal

```
5 lagarto 1 2 3  
5 papel 1 2 3 4  
5 pedra 1 2 4  
5 spock 1 3 4  
5 tesoura 1 2 3 4  
2 esmaga 2 3
```

## Orientações sobre a entrega

- (a) O trabalho é **em dupla**.
- (b) A parte de implementação do trabalho deverá ser entregue em um único arquivo compactado, com o nome dos alunos da dupla (por exemplo, Fulano\_Beltrano.zip). Indique, em um arquivo “leiam.txt” dentro do zip os nomes completos da dupla, números de matrícula, e as instruções para compilação e execução. Lembre-se que o projeto deve compilar no Linux, em qualquer máquina do laboratório.
- (c) Adicione um arquivo Makefile no seu projeto. Procurem os monitores para que os ajudem com esta parte!
- (d) Nesse zip não deve haver arquivos executáveis.
- (e) Incluir pdf da parte escrita no zip.
- (f) A data de entrega do TP é final, não sendo possível o envio com atraso. Ao final da descrição do trabalho, há outras informações disponíveis sobre a entrega.
- (g) **Data de entrega: 11/12/2022**
- (h) Valor: 30 pontos

## O que deve ser entregue:

- Documentação do trabalho em PDF. Entre outras coisas, a documentação deve conter:
  1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  2. Implementação: descrição sobre a implementação do programa.
  3. Resultados e Discussões: dados obtidos da execução do programa para diferentes casos de teste; análise crítica dos dados obtidos em comparação com o esperado.
  4. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  5. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet, se for o caso.
- Além disso, neste trabalho deve ser enviado ao professor os arquivos com todo o código-fonte. A entrega deverá ser feita via Portal Didático, seguindo as diretrizes informadas no início da descrição deste trabalho.

## Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Procurem os monitores da disciplina para tirar dúvidas!
- Clareza, indentação e comentários no programa também vão valer pontos.
- Avaliarei com maior atenção ainda a parte escrita - incluindo erros de português -. Use um revisor de texto!
- Trabalhos copiados serão penalizados com a nota zero.