

## Observações

- Em todas as questões, caso não seja possível abrir o arquivo, uma mensagem de erro deve ser escrita no **erro padrão**.
- Nas questões 1 e 3, o programa deve ser case insensitive, ou seja, não deve diferenciar maiúscula ou minúscula. Por exemplo, abacate e ABaCaTe contém as mesmas letras (não conteriam caso o programa fosse case sensitive).

## Questões

1. Escreva um programa para encontrar todas as palavras de um arquivo que contém as letras de uma dada palavra **P**.

Por exemplo, para  $P = abc$ , as palavras **abacate** e **cabana** devem ser escritas na tela (caso existam no arquivo, claro), enquanto que **acima** e **beco** não, pois a primeira não contém a letra b e a segunda não contém a letra a. O seu programa pode escrever a mesma palavra mais de uma vez, caso apareça mais de uma vez no arquivo.

**Entrada padrão:** nome de um arquivo de entrada, palavra P de até 30 caracteres (não há letra repetida)

**Saída padrão:** as palavras no arquivo que contém todas as letras de P

**Arquivo de entrada:** contém um texto qualquer

**Arquivo de saída:** -

**Erro padrão:** -

2. ▷ Implemente um programa que leia um arquivo e escreva a maior palavra que consta nesse arquivo.

**Entrada padrão:** nome de um arquivo de entrada

**Saída padrão:** a maior palavra **Arquivo de entrada:** contém um texto qualquer

**Arquivo de saída:** -

**Erro padrão:** -

3. ▷ Implemente um programa em C que descubra todas as palavras de um arquivo que contém um prefixo dado. Por exemplo, se o prefixo for **des**, o programa poderá encontrar as palavras **desterro**, **desobstruir** e **deslocamento**.

**Entrada padrão:** nome do arquivo de entrada e um prefixo **P**

**Saída padrão:** todas as palavras no arquivo de entrada que contém **P** como prefixo

**Arquivo de entrada:** contém um texto qualquer.

**Arquivo de saída:** -

**Erro padrão:** -

4. Escreva um programa que faça a leitura dos arquivos *times.txt* e *partidas.csv*. O programa deve escrever em um arquivo *classificacao.txt* o nome de cada time seguido de: quantidade de jogos e número de pontos (são 3 pontos por vitória e 1 ponto por empate).

**Entrada padrão:** -

**Saída padrão:** -

**Arquivos de entrada:** times.txt e partidas.csv

**Arquivos de saída:** classificacao.txt contendo as informações supracitadas (não precisa ordenar por pontuação)

**Erro padrão:** -

5. ▷ Implemente um programa que escreva em um arquivo todos os alunos aprovados de uma lista contida em um outro arquivo.

O arquivo de notas (disponibilizado pelo professor) contém as informações dos alunos, sendo um por linha: matrícula, nome, nota da unidade 1, nota da unidade 2 e nota da unidade 3.

**Entrada padrão:** -

**Saída padrão:** -

**Arquivo de entrada:** notas.csv

**Arquivo de saída:** aprovados.txt com a matrícula e nome de todos os alunos que obtiveram média maior ou igual a 7, um por linha

**Erro padrão:** -

6. Crie um programa com código-fonte `criptografar.c` que leia uma mensagem e escreva em um arquivo chamado `mensagemCriptografada.txt` os caracteres MSG e a mensagem de forma invertida, tal que cada caractere, diferente de espaço, é o seguinte na tabela ASCII (exceto z que é transcrito para a, Z transcrito para A e 9 transcrito para 0). Por exemplo, a mensagem "casa na ilha" é transcrita como "bimj bo btbd".

Crie também um programa com código-fonte `descriptografar.c` que abra o arquivo `mensagemCriptografada.txt` e escreva na tela a mensagem original.

Exemplo para criptografar:

| Entrada padrão | Arquivo de saída |
|----------------|------------------|
| casa na ilha   | MSGbimj bo btbd  |

Exemplo para descriptografar:

| Arquivo de entrada | Saída padrão |
|--------------------|--------------|
| MSGbimj bo btbd    | casa na ilha |

Entrada/saída para o programa criptografar:

**Entrada padrão:** mensagem de até 1000 caracteres

**Saída padrão:** -

**Arquivo de entrada:** -

**Arquivo de saída:** mensagemCriptografada.txt com a mensagem criptografada

**Erro padrão:** caso a mensagem contiver algum caractere não alfanumérico ou diferente de espaço.

Entrada/saída para o programa descriptografar:

**Entrada padrão:** -

**Saída padrão:** mensagem descriptografada

**Arquivo de entrada:** mensagemCriptografada.txt com a mensagem criptografada

**Arquivo de saída:** -

**Erro padrão:** caso o arquivo não comece com MSG