

1. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais.
2. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais e quantas são consoantes.
3. Faça um programa que receba do usuário um arquivo texto e um caractere. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.
4. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas vezes cada letra do alfabeto aparece dentro do arquivo.
5. Faça um programa que receba do usuário um arquivo texto. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas por '*'.
6. Faça um programa que leia o conteúdo de um arquivo e crie um arquivo com o mesmo conteúdo, mas com todas as letras minúsculas convertidas para maiúsculas. Os nomes dos arquivos serão fornecidos, via teclado, pelo usuário. A função que converte minúscula para maiúscula é o toupper(). Ela é aplicada em cada caractere da string.
7. Faça um programa que receba dois arquivos do usuário e crie um terceiro arquivo com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo).
8. Faça um programa para gerenciar uma agenda de contatos. Para cada contato, armazene o nome, o telefone e o aniversário (dia e mês). O programa deve permitir:
 - Inserir contato
 - Remover contato
 - Pesquisar um contato pelo nome
 - Listar todos os contatos
 - Listar os contatos cujo nome inicia com uma dada letra
 - Imprimir os aniversariantes do mês

Sempre que o programa for encerrado, os contatos devem ser armazenados em um arquivo binário. Quando o programa iniciar, os contatos devem ser inicializados com os dados contidos neste arquivo binário.

9. Crie um programa que declare uma estrutura para o cadastro de alunos.
 - Deverão ser armazenados, para cada aluno: matrícula, sobrenome (apenas um), e ano de nascimento.

- Ao início do programa, o usuário deverá informar o número de alunos que serão armazenados.
- O programa deverá alocar dinamicamente a quantidade necessária de memória para armazenar os registros dos alunos.
- O programa deverá pedir ao usuário que entre com as informações dos alunos.
- Em seguida, essas informações deverão ser gravadas em um arquivo.
- Ao final, mostrar os dados armazenados e liberar a memória alocada.

Ao iniciar o programa, forneça ao usuário uma opção para carregar os registros do arquivo para a memória do computador, alocando dinamicamente a quantidade de memória necessária.

Dica: para que o usuário possa entrar com novos dados, além dos que foram obtidos a partir do arquivo, use a função `realloc()` para realocar a quantidade de memória usada.