



(Arquivos e argumentos de linha de comando)

1. Escreva um programa, chamado **seq**, que gere uma sequência inteira de números de A a B, onde A e B são dados via linha de comando do terminal (usar **argc** e **argv**). A linha de comando deve seguir a sintaxe: **“./seq A B<ENTER>”**. A e B são números inteiros de modo que A pode ser menor ou maior que B. Assim, a sequência inteira pode ser da forma crescente (A, A+1, A+2, ..., B) ou decrescente (A, A-1, A-2, ..., B). Caso a quantidade de argumentos não esteja correta, imprima uma mensagem informando a sintaxe de entrada de comandos usada pelo programa.
2. Escreva um programa que leia a linha de comando do terminal com a sintaxe: **“./prog -start A -stop B<ENTER>”** que realiza o mesmo trabalho do exercício 1. A string **“-start”** indica o início da contagem e **“-stop”** indica o fim da contagem. As strings **“-start”** e **“-stop”** podem aparecer em qualquer ordem. (Use a biblioteca **string.h** para realizar as comparações entre *strings*).
3. Escreva um programa em linguagem C que gere N caracteres, minúsculos e aleatórios, do alfabeto e os armazene em um arquivo chamado “caracteres-randomicos”. N deve ser dado pelo usuário.
4. Escreva um programa que crie um arquivo TEXTO em disco, com o nome “dados”, e escreva neste arquivo em disco a sequência de números inteiros 1 até 100, com um número em cada linha. Abra o arquivo em um editor de texto e explique o resultado.
5. Escreva um programa que crie um arquivo BINÁRIO em disco, com o nome “dados”, e escreva neste arquivo em disco a sequência de números inteiros 1 até 100, com um número em cada linha. Abra o arquivo em um editor de texto e explique o resultado.
6. Escreva um programa que crie um arquivo BINÁRIO em disco, com o nome “abc”, e escreva a sequência de números inteiros (**int**) 1629495328, 1629495329, 1629495330. Abra o arquivo com um editor de texto e explique o resultado.
7. Escreva um programa que abra o arquivo “abc” gerado pela questão 5 e imprima em cada linha da tela o símbolo e código dos caracteres que compõem o texto.
8. Escreva um programa que gere 10 números do tipo float e salve em um arquivo em modo BINÁRIO. O mesmo programa deve abrir o arquivo salvo e carregar os 10 números em um vetor de 10 elementos e imprimir o conteúdo do vetor na tela.
9. Escreva um programa que leia, via teclado, três números complexos, organizados em uma estrutura chamada **Complex**, e os salve as estruturas em um arquivo em modo BINÁRIO.
10. Escreva um programa que leia os três números complexos salvos no arquivo da questão 9 usando a estrutura **Complex**.



11. Escreva um programa que faça o mesmo trabalho da questão 10, mas sem usar a estrutura Complex, ou seja, usando somente tipos primitivos de dados.
12. Escreva um programa que leia os dados de 3 estudantes e salve os dados em um arquivo chamado "aluno.txt". Cada estudante é representado pela estrutura Aluno, que contém: nome, número de matrícula, nota N1, N2 e N3. O mesmo programa deve escrever os dados no arquivo e recuperar os dados do mesmo arquivo antes de apresentá-los na tela.
13. Faça um programa que lê uma sequência de números inteiros em um arquivo texto e gere outro arquivo texto com a mesma sequência de modo ordenado crescente. Os nomes dos arquivos de entrada e saída devem ser passados via linha de comando (argc e argv).
14. Faça um programa que lê um arquivo texto contendo informações sobre disciplinas do curso de CC e apresente a média de aprovados e a quantidade média de alunos de cada disciplina na tela. O arquivo é organizado blocos de 4 linhas, onde a primeira contém o nome da disciplina, a segunda o nome do professor, a terceira um número inteiro que representa a quantidade de alunos matriculados e a última com um número real que representa a média de aprovados. O nome da disciplina e o nome do professor sempre terão 100 caracteres cada, excluindo o '\n'. Os caracteres vazios são preenchidos com espaços ' '. O nome do arquivo de entrada deve ser informado via linha de comando. A quantidade de linhas do arquivo é sempre múltipla de 4.
 <nome da disciplina 1>\n'
 <nome do professor da disciplina 1>\n'
 <quantidade de alunos da disciplina 1>\n'
 <média de aprovados da disciplina 1>\n'
 <nome da disciplina 2>\n'
 <nome do professor da disciplina 2>\n'
 <quantidade de alunos da disciplina 2>\n'
 <média de aprovados da disciplina 2>\n'
 ...
15. Faça um programa que cria e lê matrizes de números **float** em arquivos binários. O programa deve ler uma string que determina a opção desejada. Caso a string seja 'new' então o programa deve ler o número de linhas (nl) e colunas (nc) da matriz, o nome do arquivo de saída e gerar nl*nc números aleatórios entre 0 e 1 em uma estrutura de matriz alocada dinamicamente. Os dados devem ser escritos no arquivo de saída no modo binário. Caso a string seja 'read' o programa deve ler o nome do arquivo de entrada, carregar a matriz em uma estrutura matricial alocada dinamicamente e apresentar os valores da matriz na tela. O arquivo deve seguir a seguinte configuração: um número inteiro **nl**, um número inteiro para **nc** e **nl*nc** números **float**.