

Lista Final de Exercícios AEDS1

Essa lista não precisa ser entregue apenas, serve de guia para a avaliação final.

1. Crie uma struct Aluno com nome, matrícula e nota. Alocar dinamicamente um vetor de Aluno, preencher com N registros e salvar em alunos.txt.
2. Leia alunos.txt e grave em um novo arquivo aprovados.txt apenas os alunos com nota maior ou igual a 6.
3. Crie uma struct Produto com código, nome e preço. Cadastre N produtos usando alocação dinâmica e grave em produtos.txt.
4. Escreva um programa que leia produtos.txt, encontre o produto mais caro e exiba seus dados na tela.
5. Crie um programa que permita atualizar o preço de um produto (por código), lendo de produtos.txt e gravando em produtos_atualizados.txt.
6. Implemente um programa com Funcionario (ID, nome, salário, data). Leia N registros e salve em funcionarios.txt.
7. Leia funcionarios.txt, filtre os funcionários com salário acima de R\$ 5.000,00 e grave em diretoria.txt.
8. Escreva um programa que aloque dinamicamente um vetor de Aluno, leia os dados de alunos.txt e imprima os nomes dos alunos em ordem reversa.
9. Faça um programa que leia clientes.txt e use realloc para permitir adicionar mais clientes ao vetor, gravando o novo conjunto em clientes_atualizado.txt.
10. Leia clientes.txt, grave apenas os que possuem sobrenome "Silva" em clientes_silva.txt.
11. Crie uma função que receba um ponteiro para vetor de inteiros alocados dinamicamente e retorne a soma de todos os elementos.
12. Implemente um programa que leia N números inteiros, armazene dinamicamente e grave o maior e menor valor em resultado.txt.

13. Crie um programa que aloque um vetor de N floats, leia os dados do teclado, calcule a média e grave em media.txt.
14. Faça um programa que leia uma string de até 100 caracteres e grave em saida.txt quantos caracteres e quantas vogais ela possui.
15. Escreva uma função que receba um vetor dinâmico de inteiros e substitua cada valor pela soma dos vizinhos. Grave o vetor final em transformado.txt.
16. Crie um programa que leia dois vetores de inteiros (alocados dinamicamente), some elemento a elemento e grave o resultado em soma.txt.
17. Desenvolva um programa que leia uma matriz quadrada de ordem N (alocada dinamicamente como vetor simples), calcule a soma da diagonal principal e grave em soma_diag.txt.
18. Leia uma string e grave em palindromo.txt se ela for um palíndromo.
19. Faça um programa que leia 10 nomes, grave em nomes.txt, e depois leia esse arquivo e imprima os nomes com mais de 6 letras.
20. Implemente um vetor de Produto alocado dinamicamente, ordene por preço e grave em produtos_ordenados.txt.
21. Implemente um menu para cadastrar estudantes, gravar em estudantes.txt, listar todos e buscar por matrícula.
22. Faça um sistema de tarefas com struct Tarefa (título, status). Grave e leia de tarefas.txt, listando apenas as pendentes.
23. Crie um programa que leia alunos.txt, calcule a média geral das notas e grave esse valor em media_geral.txt.
24. Escreva um programa que leia funcionarios.txt, permita a exclusão de um funcionário por ID e salve os dados restantes em funcionarios_final.txt.
25. Desenvolva um programa que leia boletins.txt, que contém 3 notas por aluno, calcule a média e grave nome + média em medias.txt.
26. Desenvolva um programa com menu completo para um sistema de estoque: cadastrar produtos, buscar por código, atualizar preço e gerar arquivo estoque.txt.
27. Implemente um sistema de agenda de contatos (nome, telefone, email) com alocação dinâmica e gravação contínua em agenda.txt.

28. Crie um programa que leia transacoes.txt (tipo, valor, data), calcule o total por tipo (entrada/saída) e grave o resumo em resumo.txt.
29. Escreva um programa que leia clientes.txt, duplique os dados na memória, ordene por nome e grave ordenado em clientes_ordenado.txt.
30. Crie um sistema de cadastro de disciplinas onde cada linha de disciplinas.txt representa uma matéria com código, nome e professor. Permita buscas por professor.