

*Crie um arquivo para cada exercício abaixo. A ideia desse exercício é ser incremental, ou seja, o exercício 3 complementa o 2, que complementa o 1, etc.
O nome base é AlunoX.c, onde X deverá ser substituído pelo número do exercício.*

PARTE I: TIPOS DE DADOS BÁSICOS, ENTRADA E SAÍDA

1. Crie um programa que recebe os dados de um aluno de uma escola, nessa ordem: RA, primeiro nome, dia, mês e ano de matrícula e código do curso. Exiba os dados inseridos após a leitura.
Dados:
 - a) RA: Inteiro (não é igual ao da Fatec).
 - b) Primeiro nome: Texto, letras maiúsculas e minúsculas, sem símbolos.
 - c) Dia, mês, ano: Inteiros.
 - d) Código do curso: Caracter (I para Informática, L para Língua Estrangeira, A para Administração).
2. Adapte o programa 1 e permita-o receber o número de faltas (inteiro) e as notas das duas avaliações do curso (números reais, precisão simples). Novamente, exiba os dados.
3. Adapte o programa 2 para imprimir o resultado final deste aluno no formato:
Aluno **RA** – **NOME**, matriculado em **DIA/MES/ANO**
Desempenho no curso de **NOME_DO_CURSO**: **RESULTADO**
Nota final: **MEDIA_FINAL** – Presença: **PORCENTAGEM_PRESENCA**
Dados:
 - a) O resultado é apenas APROVADO ou REPROVADO.
 - b) Para o aluno ser aprovado, não pode ter mais que 10 faltas e deve ter média final maior ou igual a 6,5. O número de aulas por ano é 40 (você vai precisar disso para calcular a porcentagem de presença).
 - c) Dia, mês e ano devem ser exibidos com zeros à esquerda. Exemplo: 1 de janeiro de 2001 vai aparecer como 01/01/2001.

PARTE II: STRINGS E FUNÇÕES

4. Adapte o programa 3 para permitir receber, armazenar e imprimir o nome completo do aluno (com espaços e pontos). Cada aluno terá pelo menos duas palavras no nome.
5. No programa 4, crie funções que recebem o nome completo do aluno e retornam:
 - a) Apenas o primeiro nome do sujeito
 - b) O último nome do sujeito, em UPPER CASE (sem usar <ctype.h>).
 - c) O primeiro e o último nome, em uma mesma string. Se quiser, use as funções acima.Exiba o resultado das três funções, uma em cada linha.
6. Transfira a parte do código do programa 5 que imprime o resultado do aluno (aprovado ou reprovado, média e porcentagem de presença) para uma rotina (sem retorno).

PARTE III: REGISTROS E VETORES

7. Se ainda não fez isso (~~por alguma razão obscura na sua mente~~), com o programa 6, crie um registro (struct) para:
 - a) Os dados de cada aluno (RA, nome, data de matrícula, notas e faltas).
 - b) Datas (dia, mês e ano).

8. Em vez de receber apenas um aluno, pegue o programa 7 e crie uma lista (por vetor) para armazenar os dados de até 50 alunos. Interrompa a leitura ao receber um aluno com código igual a 0 (zero). Imprima o conteúdo da lista após a leitura.
9. Altere o programa 8 para interromper a leitura quando receber um aluno com nome igual a "FIM" (sem aspas).
10. Usando o programa 8, crie uma rotina que recebe como parâmetros o vetor de alunos e o tamanho da lista, além de três ponteiros (para inteiros). Essa rotina deve contar e retornar (através dos ponteiros informados):
 - a) Quantidade de alunos do curso de Informática
 - b) Maior nota dos alunos de Língua Estrangeira
 - c) Menor número de faltas dos alunos de AdministraçãoImprima o valor dos contadores na tela, descrevendo cada um, um em cada linha.
11. A partir do programa 10 e da rotina de contagem, adapte a rotina para uma função que copie, do vetor original de alunos, apenas os alunos aprovados em outro vetor que deverá ser retornado por referência (ponteiro). Use um ponteiro para o tamanho deste vetor de saída. Imprima a lista resultante (RA e nome apenas).

PARTE IV: ORDENAÇÃO E PESQUISA

12. Usando o programa 11, crie rotinas que implementam o método de ordenação Bolha e ordenam o vetor de alunos em:
 - a) Ordem alfabética
 - b) Ordem de média final (decrecente)

Algoritmo do bolha:

*Primeiro caso: seleciona primeiro elemento.
Compara se é maior que o próximo elemento.
Se for, trocam de lugar.
Se não for, seleciona o próximo elemento e compara.
Faz isso até não ter mais trocas.*

Imprima os dados ordenados de cada função (RA, nome e média final).
13. (Opcional) Altere o método de ordenação das rotinas do programa 12 para o método da inserção direta.

Algoritmo do método inserção direta:

*Primeiro caso: seleciona segundo elemento.
Percorre elementos anteriores.
Se achar elemento anterior menor que elemento selecionado,
insere elemento selecionado naquela posição
e "empurra" todos os elementos maiores
entre aquela posição e a posição original desse elemento
para a frente.
Seleciona o próximo elemento a partir da posição original do elemento
selecionado anteriormente.
Faz isso até percorrer todos os elementos.*
14. (Opcional) Altere as rotinas do programa 12 para ordenarem a lista por seleção direta.
 - a) Algoritmo do método seleção direta:

*Primeiro caso: Seleciona primeiro elemento.
O elemento selecionado será sempre o elemento a ser trocado.
Marca elemento selecionado como menor.
Busca elemento menor que o selecionado.
Se achar, marca como o menor.
Após percorrer todos os elementos,*

*troca elemento menor com o selecionado para troca.
 Seleciona o elemento seguinte ao selecionado e continua.
 Faz isso até o elemento selecionado ser o último.*

15. Usando o programa 12, substitua o bolha por ~~algo melhor e realmente rápido~~: uma implementação de função de comparação para a ordem alfabética e outra para a ordem decrescente de nota. Essa função será usada na ordenação do vetor através do Quick Sort (função em c: qsort).
16. Usando o programa 15, crie uma rotina que receba o vetor ordenado e um nome de aluno a ser pesquisado. Implemente uma pesquisa sequencial (percorra o vetor enquanto o contador for menor que o tamanho do vetor e o nome do aluno naquela posição for diferente do nome pesquisado). Retorne a posição do vetor caso encontre o aluno com o nome buscado, ou retorne -1 se não encontrar. Imprima a posição, se encontrado, ou imprima “NOT FOUND”.

PARTE V: ARQUIVOS

17. Usando o programa 15, altere o programa para gerar arquivos de extensão CMALP para:
 - a) Armazenar a lista de alunos ordenada por nome (arquivo “ListaAlunosNome.cmalp”)
 - b) Armazenar a lista de alunos ordenada por média (arquivo “ListaAlunosNota.cmalp”)
 Note que o programa deverá abrir e fechar corretamente o arquivo de saída e não deverá imprimir as listas na tela.
18. Adapte o programa 17 para ler o arquivo “ListaAlunosNome.cmalp” criado no exercício anterior, contar os registros (sem preencher o vetor) e imprimir:
 - a) Porcentagem de alunos de Informática em relação à quantidade total de alunos
 - b) Porcentagem de alunos de Língua Estrangeira em relação à quantidade total de alunos
 - c) Porcentagem de alunos de Administração em relação à quantidade total de alunos

PARTE FINAL: TESTE GERAL

19. (Adaptado do URI Online Judge) Crie um programa que receba dois números inteiros positivos (A e B) e imprima na tela a soma entre esses dois números (X), dada pela fórmula:

$$X = A + B$$
 O resultado impresso deve ser um único número positivo.
 Seguem abaixo exemplos de entradas e saídas. Lembrando que seu programa será testado com muito mais casos de teste do que os informados abaixo.

Entrada:	Saída:
2 2	4
5 5	10
100000 100000	200000
1024576 1	1024577