

LISTA1 - N2 - ESTRUTURA DE DADOS

Vetores, ordenação, recursividade, ponteiros, registros e estruturas de dados dinâmicas.

Prof. Ed - Ciência da Computação - Unipê

Instruções:

- Os programas **NÃO DEVEM SER COMPACTADOS**. O código-fonte deve ser enviado via upload diretamente na resposta do exercício (arquivo por arquivo)
- Cada arquivo deve ter o seguinte formato: `ED-lista2N2-questaoXX` onde `XX` é o número da questão correspondente.
- O trabalho é em **DUPLAS** (porém CADA ALUNO DEVE ENVIAR OS ARQUIVOS INDIVIDUALMENTE)
- **IMPORTANTE:** NÃO SERÃO ACEITOS TRABALHOS QUE NÃO ESTIVEREM NO FORMATO ACIMA
- **OBSERVAÇÃO:** TODOS os programas entregues devem ter o seguinte cabeçalho:

```
/*
**      Função :
**      Autor  :
**      Data   :
**      Observações:
**/
```

Onde deverá estar escrito o que o programa faz, o autor (nome, turma, a data e as observações que forem pertinentes. Os trabalhos **não serão aceitos** após a data SOB HIPÓTESE ALGUMA.

1. Criar uma base de dados usando a técnica de **lista encadeada** para gerir os **livros** de uma **biblioteca**. A informação de cada livro é: autor, título, editora, ano de edição. A base de dados deve ter as seguintes funcionalidades (funções): **Inserção ordenada; remoção;**

alteração; busca por nome do livro e busca por nome do autor. Faça um **menu** para melhorar a apresentação do programa para o usuário. (Pode usar Python ou C)

2. Criar um programa para Implementar uma **pilha de números reais**. Apresentar um **menu** para **inserir (push)**, **retirar(pop)** elementos na pilha (**exibir** a pilha a cada operação). Criar também uma função para **verificar** se a pilha está **vazia**
3. Definir uma estrutura chamada `perfil_saude` que contenha os seguintes campos:
 - Nome Completo: string com 80 caracteres
 - Sexo: um caractere
 - Data de nascimento: dia, mês e ano separados (DICA: use outra estrutura)
 - Altura: um número real
 - Peso: um número real
 -
 - Faça um programa, em C, que tenha uma função que receba esses dados e os utilize para definir os membros de uma variável `perfil_saude`. O programa deve incluir funções que calculem e retornem a idade atual do usuário em anos, a frequência cardíaca máxima e a frequência cardíaca ideal fazendo o cálculo correto caso a pessoa seja homem ou mulher, **o índice de Massa Corporal**. O programa deverá pedir a informação da pessoa, criar uma variável do tipo `perfil_saude` para ela e exibir as informações dessa variável – o que inclui TODOS os dados. (se possível, usar `getters` e `setters` para cada membro). Em seguida deverá **calcular e exibir a idade (em anos), seu IMC, e suas frequências cardíacas máxima e ideal fazendo o cálculo correto caso a pessoa seja homem ou mulher**. Deverá também exibir a tabela de valores do IMC. **Calcular também a expectativa de anos da pessoa** (quantos anos ela poderá viver ainda), baseado na expectativa de vida da população (sendo homem ou mulher – pesquisar essa informação). **Fazer um vetor de N** (informado pelo usuário) elementos com as informações do exercício acima. **Incluir o vetor nas chamadas de funções. (Usar alocação dinâmica).**
4. Implemente uma função que classifica os elementos de um vetor em ordem crescente usando o seguinte algoritmo (conhecido como “classificação por seleção”): procure pelo menor elemento no vetor e permute esse elemento com o primeiro elemento do vetor; repita este processo para o subvetor que se inicia no segundo elemento e, assim, sucessivamente; o processo termina quando o subvetor contiver apenas um elemento. Teste a função com dados gerados aleatoriamente.

5. Crie um programa para gerenciar uma pilha de processos em Python. Cada processo possui um **identificador**(número) e uma **descrição**(string). Você deverá pedir ao usuário para escolher se deseja **encerrar**, **incluir** ou **retirar** um processo da pilha. Se a operação for uma inclusão, colocar o processo na pilha e imprimir o novo estado dessa; se for uma exclusão, caso a pilha não esteja vazia, imprimir "removido o processo #identificador – descrição da pilha" e mostrar o conteúdo atual dela; se a pilha estiver vazia, mostrar "pilha vazia". Se o usuário escolher encerrar, esvazie a pilha (caso ainda existam elementos nela) e encerre o programa.
6. Faça um programa, em C, que simule uma lista de compras. As compras podem ter itens e esses itens devem ter quantidade e o nome do produto. O programa permite inserir, remover, consultar e mostrar todos os itens da lista de compras. (Usar obrigatoriamente uma lista encadeada).
7. Você foi contratado para desenvolver um sistema de gerenciamento de atendimento para uma clínica médica. O sistema deve permitir que os pacientes sejam atendidos por ordem de chegada, utilizando uma estrutura de fila.

Implemente uma struct (registro/classe) chamada `Atendimento` que possui os seguintes métodos:

1. `adicionar_paciente(paciente)` : Recebe como parâmetro o nome de um paciente e o adiciona à fila de atendimento.
2. `proximo_paciente()` : Retorna o próximo paciente a ser atendido e o remove da fila. Caso a fila esteja vazia, retorna a mensagem "Não há pacientes na fila".
3. `quantidade_pacientes()` : Retorna a quantidade de pacientes na fila de atendimento.

Instruções:

- Utilize uma estrutura de fila para gerenciar os pacientes. Você pode implementar a fila do zero ou utilizar uma biblioteca existente.
- A classe/struct/registo `Atendimento` deve ter um método `construtor` para inicializar a fila de atendimento vazia.
- Utilize a linguagem de programação de sua preferência (C ou Python)