

As questões da parte teórica devem ser respondidas sem consulta a qualquer material. Crie um arquivo de texto com seu nome para informar as respostas. Sugestão: Utilize o LibreOffice Writer para editar as respostas e gerar uma versão “pdf” para entrega (Arquivo -> Exportar como PDF). Não é necessário copiar as questões para o editor de textos.

1. (0,4) Quais são as etapas fundamentais do algoritmo “Selection Sort”?
  - a) Dividir e conquistar.
  - b) Inserir e organizar.
  - c) Mesclar e dividir.
  - d) Selecionar e trocar.
  - e) Selecionar e comparar.
2. (0,4) Qual é o principal objetivo dos ponteiros em programação?
  - a) Armazenar valores constantes.
  - b) Armazenar valores maiores.
  - c) Armazenar valores temporários.
  - d) Evitar erros de compilação.
  - e) Manipular e acessar a memória.
3. (0,4) O que é alocação dinâmica de memória em programação?
  - a) Alocação de memória automaticamente pelo sistema operacional.
  - b) Alocação de memória durante a compilação.
  - c) Alocação de memória durante a execução do programa.
  - d) Alocação de memória em tempo de design.
  - e) Alocação de memória em tempo de debug.
4. (0,4) O que acontece se você tentar acessar um local de memória não alocado por meio de um ponteiro?
  - a) O ponteiro se tornará nulo.
  - b) O programa irá travar.
  - c) O programa irá continuar normalmente.
  - d) Um erro de compilação irá ocorrer.
  - e) Um erro de execução irá ocorrer.
5. (0,4) O que é um ponteiro NULL?
  - a) Um ponteiro que aponta para uma posição de memória específica.
  - b) Um ponteiro que aponta para um local inválido.
  - c) Um ponteiro que contém o valor zero.
  - d) Um ponteiro que aponta para o início da memória.
  - e) Um ponteiro que aponta para o fim da memória.
6. (0,4) Qual é a principal desvantagem da alocação dinâmica de memória em comparação com a alocação estática?
  - a) A alocação dinâmica é mais difícil de se implementar.
  - b) A alocação dinâmica consome mais recursos do sistema.
  - c) A alocação dinâmica pode levar a vazamentos (programa não libera memória que já não é utilizada) de memória se não for gerenciada corretamente.
  - d) A alocação dinâmica não pode ser usada para criar estruturas de dados complexas.

e) Não há nenhuma desvantagem na alocação dinâmica de memória.

7. (0,4) O que é uma "struct" em programação?

- a) Um tipo de dado que armazena várias variáveis de tipos diferentes.
- b) Um tipo de dado que armazena várias variáveis de mesmo tipo.
- c) Uma função que retorna um valor.
- d) Uma variável que pode armazenar múltiplos valores.
- e) Um ponteiro para uma variável.

8. (0,4) Qual a principal diferença entre uma lista sequencial e uma lista encadeada?

- a) Uma lista sequencial ocupa menos espaço de memória.
- b) Uma lista encadeada não exige espaço contínuo na memória para armazenar os elementos.
- c) Uma lista encadeada permite acesso aleatório aos elementos.
- d) Uma lista sequencial é mais eficiente para operações de inserção e remoção.
- e) Não há diferença significativa entre elas.

9. (0,4) O que é um Tipo Abstrato de Dados (TAD)?

- a) Uma estrutura de dados que define operações e seus comportamentos, sem especificar detalhes de implementação.
- b) Um tipo de dado que só pode armazenar números inteiros.
- c) Uma função que não retorna valor.
- d) Um tipo de dado que pode armazenar tanto números inteiros quanto números de ponto flutuante.
- e) Uma estrutura de dados que armazena apenas caracteres.

10. (0,4) O que acontece quando tentamos remover um elemento de uma fila vazia?

- a) Nada acontece e a fila permanece vazia.
- b) O programa trava.
- c) O último elemento da fila é removido.
- d) O primeiro elemento da fila é removido.
- e) Um erro de execução ocorre.

Você pode utilizar os [materiais e consultas a websites](#) para auxiliar na resolução dos exercícios práticos. Crie uma pasta para cada exercício (exemplo, “ex\_11” e “ex\_12”). Para entrega, realize a compactação das pastas em um único arquivo.

**Importante:** É proibida a utilização de qualquer IA (Inteligência Artificial) para auxílio na resolução dos exercícios.

11. (2,0) Implemente um algoritmo para ordenação, em ordem crescente, da lista a seguir:

41	23	16	2	52	39	9	16
----	----	----	---	----	----	---	----

Imprima o vetor resultante para cada um dos passos do processo de ordenação e a quantidade de iterações necessárias para realizá-lo.

12. (4,0) Elabore um programa para o controle da organização de malas em uma viagem. Cada mala possui um localizador único (um código), o nome do passageiro e o peso da mala (5kg ou 10kg). Existem duas filas para a entrega das malas. Clientes preferenciais utilizam a fila prioritária e terão suas malas devolvidas antes dos demais passageiros no destino final. Clientes não preferenciais utilizam a fila comum e terão suas malas devolvidas após os prioritários. Cada fila permite o acesso a 10 compartimentos. Se o compartimento prioritário estiver cheio, a mala deverá ser incluída no compartimento da fila não prioritária e o cliente (preferencial) receberá um bônus de R\$ 100,00 pelo inconveniente. Se o compartimento não prioritário também estiver cheio, clientes preferenciais recebem um bônus adicional de R\$ 100,00 e terão de despachar a mala. Clientes não preferencias não possuem direito a bônus caso o compartimento esteja cheio e terão que despachar a mala.

Devem ser criadas as funcionalidades para organização das malas nos dois compartimentos (preferencial e não preferencial) e impressão do processo de entrega das malas de acordo com a prioridade, exibindo o nome do passageiro, código da mala, status (preferencial ou não), peso da mala e bônus, caso este tenha sido gerado.

*Observação 1: Cada cliente pode despachar uma única mala.*

*Observação 2: Cada viagem pode levar até 30 passageiros.*

*Observação 3: As malas despachadas serão entregues por último.*

*Utilize struct e TAD.*