

## **Alocação Dinâmica:**

1. O que é alocação dinâmica de memória?

Alocação dinâmica de memória é a forma de utilizar a memória RAM do computador de forma dinâmica, ou seja, conforme é solicitado pelo programa e não alocada previamente com uma variável carregada

2. Qual é a principal função da alocação dinâmica em linguagens de programação?

Permitir alocar e liberar memória conforme necessário durante a execução do programa

3. Em quais situações é recomendado usar alocação dinâmica?

Ao não saber exatamente a quantidade de memória necessária para determinada variável

4. Quais são os principais operadores utilizados para alocar e liberar memória dinamicamente em C?

Malloc, calloc, realloc e free

5. O que acontece se você esquecer de liberar a memória alocada dinamicamente após o uso?

O espaço alocado não é liberado para uso para o próprio programa ou para outros programas no computador, ocasionando lentidão e talvez travamentos

6. Cite um exemplo prático de alocação dinâmica em C.

Pode ser utilizada em uma lista

7. Quais são os riscos associados à alocação dinâmica inadequada?

Travamentos e lentidão do programa e da máquina

8. Como a função `malloc()` difere da função `calloc()`?

Calloc alocará a memória resultante da operação de multiplicação entre a quantidade de elementos a serem alocados e o tamanho em byte de cada elemento

Malloc trabalha diferente, alocando apenas a quantidade de bytes necessária

além disso malloc inicia-se com null enquanto calloc inicia-se com 0

9. O que é fragmentação de memória e como a alocação dinâmica pode contribuir para esse problema?

Fragmentação é a perda de espaço em um tamanho definido, por exemplo, ao declarar que uma variável vai utilizar 4 bytes e ela utiliza apenas 3, temos 1byte de espaço fragmentado e não utilizado que poderia ser usado em outra variável. A alocação dinâmica soluciona esse problema alocando diretamente a quantidade em bytes à serem utilizadas pela memória, como por exemplo no uso do malloc com o `sizeof`

10. Qual é o papel da função `free()` no processo de alocação dinâmica?

Devolver a área de memória alocada pelas demais funções para que outros programas possam utilizar

## Listas Encadeadas:

1. O que é uma lista encadeada?

É uma sequência de objetos do mesmo tipo alocados na memória

2. Quais são as partes principais que compõem um nó em uma lista encadeada simples?

O valor ou elemento e o ponteiro para o próximo item da lista

3. Qual é a diferença entre uma lista encadeada simples e uma lista encadeada dupla?

A lista encadeada simples apresenta somente o parâmetro next, indicando a posição do próximo item na lista, na encadeada dupla, temos o parâmetro previous que indica a posição do item anterior

4. Quais são as vantagens e desvantagens de usar listas encadeadas em relação a arrays estáticos?

Vantagens: crescimento dinâmico e inserção/remoção eficiente

Desvantagens: maior consumo de memória devido aos ponteiros e acesso sequencial

5. Como é possível inserir um novo elemento no início de uma lista encadeada?

Você deve identificar se existe um primeiro elemento na lista, caso não exista é só fazer o input dos dados, caso exista, deve fazer um laço indicando a posição será em NULL e o parametro next será do item que estava primeiro, iterando sobre todos os itens na lista alterando o parametro next para o item posterior do atual

6. Explique o conceito de "ponteiro de nó cabeça" em uma lista encadeada.

É um ponteiro presente no ultimo item da lista que aponta para o primeiro item na lista encadeada

7. Descreva o processo de remoção de um elemento específico de uma lista encadeada.

Primeiro precisamos de um identificador único o elemento na lista, por exemplo, se usarmos uma struct de funcionário devemos pesquisar pelo nome ou cpf do mesmo na lista, após encontra-lo usamos o metodo free para eliminar o item da lista e em seguida alocamos o item posterior no lugar do item atual através do parametro next, alterando também todos os itens posteriores

8. Como você percorreria uma lista encadeada para imprimir seus elementos?

Através de um laço, podendo ser um while neste caso e enquanto o parâmetro next ou aux, dependendo da sua forma de chamar parâmetro que identifica o próximo elemento da lista, for diferente de null ele imprimira o item da lista

9. O que é uma lista circular e como ela se diferencia de uma lista encadeada comum?

A lista circular possui dois elementos chaves, o primeiro é um ponteiro que indica a posição do primeiro item (first), e a segunda que o último item da lista deve apontar para o primeiro item a partir do "first"

10. Quais são os cenários em que as listas encadeadas são preferíveis sobre outras estruturas de dados?

Quando se faz necessário armazenar diversos itens como se fossem um vetor mas não se sabe ao certo quantos inputs serão necessários, por exemplo, em uma criação de struct de funcionários, temos diversos itens dentro do funcionário e não sabemos ao certo quantos funcionários serão criados, criamos a lista para ser possível fazer uma entrada de um novo funcionário ou remoção de um antigo, também é possível alterar dados

### **Listas Duplamente Encadeadas:**

1. O que é uma lista duplamente encadeada e como ela se diferencia de uma lista encadeada simples?

Uma sequência de elementos que possuem um ponteiro para o próximo item da lista, diferente da lista simples, a duplamente encadeada apresenta um ponteiro para o item anterior da lista

2. Quais são as vantagens de usar listas duplamente encadeadas em relação a listas encadeadas simples?

Pode ser aplicados algoritmos de busca como por exemplo o da busca binária

3. Quais são as partes principais que compõem um nó em uma lista duplamente encadeada?  
O valor do elemento, ponteiro para o próximo item da lista e um ponteiro para o item anterior da lista

4. Como você insere um novo elemento no final de uma lista duplamente encadeada?

5. Para inserir um novo elemento no final de uma lista duplamente encadeada, basta criar um novo nó, ajustar os ponteiros corretamente para incluir o novo nó na sequência da lista e atualizar o ponteiro do final da lista para apontar para o novo nó

6. Descreva o processo de remoção de um elemento específico de uma lista duplamente encadeada.

Após encontrar o item através de um identificador único liberamos o espaço da memória que e atualizamos os valores dos ponteiros próximo e anterior dos itens seguintes na lista

7. Quais são as operações necessárias para inverter a ordem dos elementos em uma lista duplamente encadeada?

Identificaremos o primeiro e último item da lista, após isso com uma estrutura auxiliar trocaremos os dois de posição, então com um laço partindo do novo primeiro elemento atualizaremos os valores do próximo item da lista com os valores do item anterior ao nosso novo último elemento, para isso também utilizaremos a estrutura auxiliar para armazenar temporariamente o valor e não perder informações

8. Como você imprimiria os elementos de uma lista duplamente encadeada na ordem inversa?

A partir de um laço identificaria o último elemento com o ponteiro do próximo apontando para null, o iterador do laço seguiria o ponteiro do item anterior da lista imprimindo os valores até encontrar o elemento head

9. Como lida com a manipulação de ponteiros em uma lista duplamente encadeada?

Usando um elemento auxiliar para armazenar temporariamente os valores

10. Quais são as desvantagens de utilizar listas duplamente encadeadas em comparação com listas encadeadas simples?

Ocupam mais espaço em memória devido a informação extra do previous

11. Cite alguns exemplos de situações em que listas duplamente encadeadas são particularmente úteis.

Em programas que não sabemos ao certo quantos itens serão necessários armazenar e também precisaremos realizar buscas nos elementos além da criação e deleção