



# ESTRUTURA DE DADOS – 03A – 2022.2

[Página inicial](#)

[Meus cursos](#)

[ESTRUTURA DE DADOS – 03A – 2022.2](#)

[Tópico 4. Ponteiros e Alocação Dinâmica de Memória](#)

[Exercícios – Ponteiros e alocação dinâmica](#)

## Exercícios – Ponteiros e alocação dinâmica

**Questão 01.** Faça um programa que leia um valor  $N$  e crie dinamicamente um vetor de  $N$  elementos e passe esse vetor para uma função que vai ler os elementos desse vetor. Depois, no programa principal, o vetor preenchido deve ser impresso. Além disso, antes de finalizar o programa, deve-se liberar a área de memória alocada.

**Questão 02.** Faça uma função que receba um valor  $N$  e crie dinamicamente um vetor de  $N$  elementos e retorne um ponteiro. Crie uma função que receba um ponteiro para um vetor e um valor  $N$  e imprima os  $N$  elementos desse vetor. Construa também uma função que receba um ponteiro para um vetor e libere esta área de memória. Ao final, crie uma função principal que leia um valor  $N$  e chame a função criada acima. Depois, a função principal deve ler os  $N$  elementos desse vetor. Então, a função principal deve chamar a função de impressão dos  $N$  elementos do vetor criado e, finalmente, liberar a memória alocada através da função criada para liberação.

**Questão 03.** Crie uma estrutura chamada Cadastro. Essa estrutura deve conter o nome, a idade e o endereço de uma pessoa. Agora, escreva uma função que receba um inteiro positivo  $N$  e retorne o ponteiro para um vetor de tamanho  $N$ , alocado dinamicamente, dessa estrutura. Solicite também que o usuário digite os dados desse vetor dentro da função.

**Questão 04.** Faça um programa que leia um valor inteiro  $N$  não negativo. Se o valor de  $N$  for inválido, o usuário deverá digitar outro até que ele seja válido (ou seja, positivo). Em seguida, leia um vetor  $V$  contendo  $N$  posições de inteiros, em que cada valor deverá ser maior ou igual a 2. Esse vetor deverá ser alocado dinamicamente.

**Questão 05.** Escreva um programa que aloque dinamicamente uma matriz de inteiros. As dimensões da matriz deverão ser lidas do usuário. Em seguida, escreva uma função que receba um valor e retorne 1, caso o valor esteja na matriz, ou retorne 0, no caso contrário.

**Questão 06.** Crie uma função que receba uma string (um vetor de caracteres cujo último caractere é `'\0'`) e retorne o ponteiro para essa string invertida.

**Questão 07.** Escreva uma função que receba um valor inteiro positivo  $N$  por parâmetro e retorne o ponteiro para um vetor de tamanho  $N$  alocado dinamicamente. Esse vetor deverá ter os seus elementos preenchidos com certo valor, também passado por parâmetro. Se  $N$  for negativo ou igual a zero, um ponteiro nulo deverá ser retornado.

**Questão 08.** Escreva uma função que receba como parâmetro um valor  $N$  e retorne o ponteiro para uma matriz alocada dinamicamente contendo  $N$  linhas e  $N$  colunas. Essa matriz deve conter o valor 1 na diagonal principal e 0 nas demais posições.

**Questão 09.** Escreva uma função que receba como parâmetro dois vetores,  $A$  e  $B$ , de tamanho  $N$  cada. A função deve retornar o ponteiro para um vetor  $C$  de tamanho  $N$  alocado dinamicamente, em que  $C[i] = A[i] + B[i]$ .

**Questão 10.** Escreva uma função que receba como parâmetro duas matrizes,  $A$  e  $B$ , e seus tamanhos. A função deve retornar o ponteiro para uma matriz  $C$ , em que  $C$  é o produto da multiplicação da matriz  $A$  pela matriz  $B$ . Se a multiplicação das matrizes não for possível, retorne um ponteiro nulo.

**Questão 11.** Construa um programa (*main*) que aloque em tempo de execução (dinamicamente) uma matriz de ordem  $M \times N$  (linha por coluna), usando  $1 + M$  chamadas a função *malloc*. Agora, aproveite este programa para construir uma função que recebendo os parâmetros  $M$  e  $N$  aloque uma matriz de ordem  $M \times N$  e retorne um ponteiro para esta matriz alocada. Crie ainda uma função para liberar a área de memória alocada pela matriz. Finalmente, crie um novo programa (*main*) que teste/use as duas funções criadas acima.

**Questão 12.** Criar uma estrutura que represente uma pessoa, contendo nome, data de nascimento e CPF. Crie uma variável que é um ponteiro para esta estrutura (no programa principal). Depois crie uma função que receba este ponteiro e preencha os dados da estrutura e também uma função que receba este ponteiro e imprima



os dados da estrutura. Finalmente, faça a chamada a esta função na função principal.

**Questão 13.** Crie um programa que declare uma estrutura (registro) para o cadastro de alunos.

- (a) Deverão ser armazenados, para cada aluno: matrícula, sobrenome (apenas um) e ano de nascimento.
- (b) Ao início do programa, o usuário deverá informar o número de alunos que serão armazenados
- (c) O programa deverá alocar dinamicamente a quantidade necessária de memória para armazenar os registros dos alunos.
- (d) O programa deverá pedir ao usuário que entre com as informações dos alunos.
- (e) Ao final, mostrar os dados armazenados e liberar a memória alocada.

**Questão 14.** Considere um cadastro de produtos de um estoque, com as seguintes informações para cada produto:

- **Código de identificação do produto:** representado por um valor inteiro
- **Nome do produto:** com até 50 caracteres
- **Quantidade disponível no estoque:** representado por um número inteiro
- **Preço de venda:** representado por um valor real

- (a) Defina uma estrutura, denominada produto, que tenha os campos apropriados para guardar as informações de um produto
- (b) Crie um conjunto de N produtos (N é um valor fornecido pelo usuário) e peça ao usuário para entrar com as informações de cada produto
- (c) Encontre o produto com o maior preço de venda
- (d) Encontre o produto com a maior quantidade disponível no estoque

**Questão 15.** Faça um programa que leia números do teclado e os armazene em um vetor alocado dinamicamente. O usuário irá digitar uma sequência de números, sem limite de quantidade. Os números serão digitados um a um e, sendo que caso ele deseje encerrar a entrada de dados, ele irá digitar o número ZERO. Os dados devem ser armazenados na memória deste modo:

- Inicie com um vetor de tamanho 10 alocado dinamicamente;
- Após, caso o vetor alocado esteja cheio, aloque um novo vetor do tamanho do vetor anterior adicionado espaço para mais 10 valores (tamanho N+10, onde N inicia com 10);
- Copie os valores já digitados da área inicial para esta área maior e libere a memória da área inicial;
- Repita este procedimento de expandir dinamicamente com mais 10 valores o vetor alocado cada vez que o mesmo estiver cheio. Assim o vetor irá ser 'expandido' de 10 em 10 valores. Ao final, exiba o vetor lido.

**Questão 16.** Faça um programa para associar nomes às linhas de uma matriz de caracteres. O usuário irá informar o número máximo de nomes que poderão ser armazenados. Cada nome poderá ter até 30 caracteres com o '\0'. O usuário poderá usar 5 opções diferentes para manipular a matriz:

1. Gravar um nome em uma linha da matriz;
2. Apagar o nome contido em uma linha da matriz;
3. Informar um nome, procurar a linha onde ele se encontra e substituir por outro nome;
4. Informar um nome, procurar a linha onde ele se encontra e apagar;
5. Pedir para recuperar o nome contido em uma linha da matriz;

**Questão 17.** Faça um programa que:

- (a) Peça para o usuário entrar com o nome e a posição (coordenadas X e Y) de N cidades e as armazene em um vetor de estruturas (N é informado pelo usuário);
- (b) Crie uma matriz de distâncias entre cidades de tamanho N x N;
- (c) Calcule as distâncias entre cada duas cidades e armazene na matriz;
- (d) Exiba na tela a matriz de distâncias obtida;
- (e) Quando o usuário digitar o número de duas cidades o programa deverá retornar a distância entre elas.

**Questão 18.** Faça um programa que leia quatro números a,b,c,d, que serão as dimensões de duas matrizes, e:

- Crie e leia uma matriz, dadas as dimensões dela;
- Crie e construa uma matriz que seja o produto de duas matrizes. Na sua função **main()**, imprima as duas matrizes e o produto entre elas, se existir.

Última atualização: sexta, 21 mai 2021, 19:24





©2020 – Universidade Federal do Ceará – Campus Quixadá.

Todos os direitos reservados.

Av. José de Freitas Queiroz, 5003

Cedro – Quixadá – Ceará CEP: 63902-580

Secretaria do Campus: (88) 3411-9422

 Obter o aplicativo para dispositivos móveis

