

Lista de Exercícios da 3ª Unidade

(Ponteiros, Alocação dinâmica, Estruturas)

1. Construa um programa em C que realize as seguintes operações:

a) Faça uma função `int* AlocaMemoria(int n)` que receba um valor `n` como parâmetro e crie dinamicamente um vetor de `n` elementos inteiros e retorne um ponteiro para o início deste vetor;

b) Crie uma função `void imprime(int * p, int n)` que receba um ponteiro para um vetor e um valor `n` e imprima os `n` elementos desse vetor;

c) Construa uma função `void LiberaMemoria(int* p)` que receba um ponteiro para um vetor e libere (usando `free()`) esta área de memória;

d) Crie a função principal que leia um valor `n` do usuário e chame a função `AlocaMemoria`. Depois, a função principal deve ler os `n` elementos desse vetor. Então, a função principal deve chamar a função `Imprime` para impressão dos `n` elementos do vetor criado e, finalmente, liberar a memória alocada através da função `LiberaMemoria`.

2. Implemente uma função que recebe 2 vetores de valores reais em R3 (ou seja, vetores de tamanho 3) e retorne um ponteiro que corresponde ao produto vetorial entre esses vetores. A função principal deve imprimir na tela do usuário as coordenadas do produto vetorial obtido.

Dica: utilize alocação de memória para que seu ponteiro de retorno se comporte como o vetor que armazenará o resultado do produto vetorial.

3. Crie 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. Na função principal pergunte ao usuário o número `n` de pessoas a serem representadas e utilize "`n`" para alocar a memória para seu ponteiro de estrutura. Ao final das chamadas das funções, não esqueça de liberar a memória alocada!

4. Faça um programa que leia do teclado os dados de `n` alunos. Utilize vetor de registro para armazenar os dados de cada aluno. Esse vetor deve ser obtido através de uma alocação dinâmica, a partir do valor de `n` informado pelo usuário. O registro contém CPF (tipo inteiro), NOME (tipo String) e um vetor de reais de tamanho 3 chamado NOTAS. Ao final, imprima seus dados, seguido da média das três notas de cada aluno e sua situação (aprovado – média ≥ 7 , reprovado – média < 3 , exame final – média entre 3 e 6);

5. Faça um programa que:

a) Lê os dados de um estoque de loja. Cada produto do estoque possui um CODIGO (inteiro), uma QUANTIDADE (inteiro) e um PRECO (real). Armazene estes dados em um vetor de registros de 5 posições.

b) Crie um novo vetor de registros e copie apenas aqueles produtos digitados no item a) que custam acima de R\$ 10,00.

- c) Imprima a média de preço dos produtos que custam acima de R\$ 10,00.
- d) Imprima todos os produtos que custam mais de R\$ 10,00.

6. Escreva um programa que crie um pequeno cadastro de veículos para aluguel em uma locadora de veículos em um vetor de registros com a seguinte estrutura:

- Marca: (string de 15 posições)
- Modelo: (string de 20 posições)
- Valor da diária: (real)
- Número de portas: (integer)

Após a criação do cadastro, o programa deve, em processo repetitivo, ler do teclado o número de portas e o valor máximo da diária do veículo desejado e, em seguida, apresentar na tela uma relação contendo a marca e o modelo de todos os veículos cadastrados e que satisfaçam as condições solicitadas. O processo repetitivo termina quando for informado o valor 0 (zero) para o número de portas.

OBS.:

1) O número de veículos a serem cadastrados deve ser lido no início do programa no intervalo [1,20];

Enquanto não for digitado um número no intervalo definido o programa não deve prosseguir.

2) O programa deve conter necessariamente:

a) Uma função que receba o número de veículos a serem cadastrados e retorne por parâmetro o vetor registro com os veículos cadastrados;

b) Uma função ou procedimento para imprimir uma relação contendo a marca e modelo de todos os veículos que tenham as características indicadas a ser chamada no programa principal.