INSTITUTO FEDERAL
GOIANO
Câmpus Rio Verde

Exercícios: Alocação Dinâmica

1. Faça um programa que leia do usuário o tamanho de um vetor a ser lido, faça a alocação dinâmica de memória, leia do usuário seus valores inteiros e imprima o vetor lido.
2. Faça um programa que leia uma quantidade qualquer de números armazenando-os na memória, pare a leitura quando o usuário entrar um número negativo e imprima o vetor lido.
3. Crie um programa que:
   (a) Aloque dinamicamente um array de 5 números inteiros,
   (b) Peça para o usuário digitar os 5 números no espaço alocado,
   (c) Mostre na tela os 5 números,
   (d) Libere a memória alocada.
4. Faça um programa que leia n inteiros (def nidos pelo usuário) armazenando-os em uma memória alocada dinamicamente. Em seguida, mostre quantos dos n números são pares e quantos são ímpares.
5. 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 f nal, mostrar os dados armazenados e liberar a memória alocada.
6. Faça um programa em C que receba do usuário o tamanho de uma string e chame uma função para alocar dinamicamente essa string. Em seguida, o usuário deverá informar o conteúdo dessa string. O programa imprime a string sem suas vogais.
7. Faça um programa que:
   (a) Crie uma matriz de distâncias entre n cidades diferentes,
   (b) Peça para o usuário entrar com as distâncias entre as cidades
   (c) Exiba na tela a matriz de distâncias criada
   (d) Quando o usuário digitar o número de duas cidades o programa deverá retornar a distância entre elas
8. Escreva um programa que aloque dinamicamente uma matriz (de inteiros) de dimensões def nidas pelo usuário. Em seguida, implemente uma função que receba um valor, retorne 1 caso o valor esteja na matriz ou retorne 0 caso não esteja na matriz.

9. Construa um programa que leia da entrada padrão o número de linhas e de colunas de uma matriz de números reais, aloque espaço dinamicamente para esta e a inicialize, com valores fornecidos pelo usuário, através da entrada padrão. Ao final o programa deve retornar a matriz na saída padrão com layout apropriado.

10. Faça um programa que leia um número n e:

    - Crie e leia um vetor de inteiro de n posições;
    - Conte os múltiplos de um número inteiro x num vetor e mostre-os na tela.

    Na sua função main(), mostre quantos múltiplos foram encontrados.

11. Faça um programa que leia dois números n e m e:

    - Crie e leia uma matriz de inteiros n x m. Use ponteiro simples;
    - Localize os três maiores números de uma matriz e mostre a linha e coluna onde estão.

12. Faça um programa que leia dois números n e m e, usando ponteiros duplos:

    - Crie e leia uma matriz n x m de inteiros;
    - Crie e construa uma matriz transposta m x n de inteiros.

    Na sua função main(), mostre as duas matrizes.
    Dica: lembre-se de que uma matriz é um 'vetor de vetores'.

13. Escreva um trecho de código em 'C' para fazer a alocação dinâmica (calloc ou malloc) dos blocos de dados conforme solicitado abaixo:

    (a) Vetor de 1024 Bytes (1Kbyte).
    (b) Tabela de Inteiros de dimensão 10x10.
    (c) Tabela para armazenar um vetor de 50 registros contendo: nome do produto (30 caracteres), código do produto (inteiro) e preço em reais.
    (d) Texto de até 100 linhas com até 80 caracteres em cada linha.

14. Faça um programa para armazenar em memória um vetor de dados contendo 1500 valores do tipo int, usando a função de alocação dinâmica de memória CALLOC:

    (a) Faça um loop e verifique se o vetor contém realmente os 1500 valores inicializados com zero (conte os 1500 zeros do vetor).
    (b) Atribua para cada elemento do vetor o valor do seu índice junto a este vetor.
    (c) Faça um loop e conte quantos dos 1500 valores são diferentes de zero no vetor. Exiba na tela o valor final da contagem.
    (d) Exibir na tela os 10 primeiros e os 10 últimos elementos do vetor.

15. Faça um programa que pergunte ao usuário quantos valores ele deseja armazenar em um vetor de doubles, depois use a função MALLOC para reservar (alocar) o espaço de memória de acordo com o especificado pelo usuário. Use este vetor dinâmico como um vetor comum, atribuindo aos 10 primeiros elementos do vetor valores aleatórios (rand) entre 0 e 100. Exiba na tela os valores armazenados nos 10 primeiros elementos do vetor (O vetor deve ter pelo menos um tamanho igual a 10 doubles, ou mais).

16. Faça um laço de entrada de dados, onde o usuário deve digitar uma sequência de números, sem limite de quantidade de dados a ser fornecida. O usuário irá digitar os números um a um, sendo que caso ele deseje encerrar a entrada de dados, ele ira digitar o número Zero. No final, todos os dados digitados deverão ser salvos em um arquivo texto em disco. Atenção: os dados devem ser armazenados na memória deste modo... faça com que o programa inicie criando um ponteiro para um bloco (vetor) de 10 valores inteiros, e alocando dinamicamente espaço em memória para este bloco; após, caso o vetor alocado esteja cheio; aloque um novo vetor do tamanho do vetor anterior adicionado com 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.

17. Faça um programa que simule 'virtualmente' a memória de um computador: o usuário começa especificando o tamanho da memória (define quantos bytes tem a memória), e depois ele irá ter 2 opções: inserir um dado em um determinado endereço, ou consultar o dado contido em um determinado endereço. A memória deve iniciar com todos os dados zerados.

18. Baseado no programa anterior, implemente um mecanismo para associar nomes as posições de memória (um nome de uma posição de memória tem até 10 caracteres, com o `'\0'`). O usuário irá poder usar 5 opções diferentes para manipular a memória: 1) Associar um nome com uma posição de memória; 2) Informar um endereço e um valor para armazenar neste endereço; 3) Informar um nome de uma posição de memória e armazenar um valor nesta posição; 4) Pedir para recuperar o dado contido em uma posição de memória; 5) Pedir para recuperar o dado, indicando o nome da posição de memória onde ele se encontra.

19. 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 ate 50 caracteres
    - Quantidade disponível no estoque: representado por um número inteiro
    - Preço de venda: representado por um valor real

    (a) Defina uma estrutura em C, 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

20. Escreva um programa que lê primeiro os 6 números gerados pela loteria na noite de sábado na TV e depois Lê seus próprios 6 números. Então, o programa compara quantos números o jogador acertou. Em seguida, ele aloca espaço para um vetor de tamanho igual a quantidade de números corretos e guarda os números corretos nesse vetor. Finalmente, o programa exibe os números sorteados (7 números) e os seus números corretos.