

Exercícios
Alocação dinâmica, vetores e matrizes

1. Implemente um programa que utilize uma estrutura para armazenar um cadastro de estudantes, conforme detalhado abaixo. Este cadastro deverá ser armazenado em um vetor, sendo que o número de posições será indicado via teclado. Os dados dos estudantes também deverá ser digitado pelo usuário. Após a atribuição dos valores, o programa deve imprimir (na tela) as posições e seus conteúdos.

```
typedef struct{  
    char nome[50];  
    char endereco[100];  
    int matricula;  
} estudante;
```

Exemplo de saída:

Estudante 1:
Nome: <nome>
Endereço: <endereço>
Matrícula: <matricula>

Estudante 2:
Nome: <nome>.....

2. Implemente um programa que receba números inteiros do usuário indefinidamente, armazenando-os em um vetor. O programa finaliza quando o usuário entrar com o número 0. Regras:

- a) Aloque, inicialmente, memória para 5 inteiros;
- b) Caso o usuário entrar com mais inteiros, faça a realocação, alocando espaço para mais 5 inteiros e assim sucessivamente;

3. Implemente um programa que leia um número de ordem de uma matriz, faça sua alocação e leia os dados que irão compô-la. Em seguida, implemente uma função que indique se essa matriz de números inteiros é uma matriz identidade ou não. A função deve retornar 1 se a matriz for uma matriz identidade, e 0 caso contrário. A função recebe como parâmetros a matriz de inteiros, usando a representação de matrizes através de **vetores simples**, e um inteiro n, indicando a dimensão da matriz. Essa função deve obedecer o protótipo: `int matriz_identidade(int* mat, int n)`.

A matriz identidade é uma matriz quadrada (mesmo número de linhas e colunas) que possui na sua diagonal principal apenas o valor 1, e os demais elementos com valor 0. Veja abaixo um exemplo de uma matriz identidade de ordem 4:

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4. Implemente o mesmo programa do exercício 4, porém utilize a representação de matrizes através de **vetores de vetores**. A função que indicará se a matriz é uma matriz identidade deve obedecer o protótipo: `int matriz_identidade(int** mat, int n)`.