

1 - Criar um programa que declare uma variável chamada num do tipo int com o valor 10. Crie um ponteiro que aponte para a variável num. Crie uma função com tipo de retorno **void** que recebe o endereço da variável num e altere o valor para 15 através do uso de ponteiros. Na função principal mostre o valor da variável num.

2 - Implemente uma função que receba como parâmetro um vetor de números inteiros e seu tamanho e retorne quantos números pares estão armazenados no vetor.

A função deve obedecer ao protótipo:

int pares (int tamanho, int* vet);

3 - Criar uma função para alocar um vetor de inteiros utilizando ponteiros e alocação dinâmica (**malloc**). A função tem o seguinte protótipo:

int* alocaVetor (int tamanho);

4 - Implementar as funções abaixo para alocar dinamicamente uma matriz e liberar sua memória.

float alocaMatriz(int linhas, int colunas);**

void liberaMatriz(float **matriz);

5 - Usando as funções dos exercícios anteriores faça:

- Leia do usuário o número de elementos a serem armazenados no vetor.
- Crie um vetor com o tamanho especificado pelo usuário.
- Leia do usuário o número de linhas da matriz. O número de colunas será o mesmo número de elementos do vetor lido anteriormente.
- Crie a matriz com os valores entrados pelo usuário.
- Preencha o vetor com valores inteiros. (Aleatórios ou lidos do usuário).
- Preencha cada linha da matriz com os valores do vetor multiplicados pelo número da linha + 0.5.
- Imprima ao final todos os elementos do vetor e da matriz.
- Libere a memória utilizada pelo vetor e pela matriz

Exemplo de execução:

Nº de elementos do vetor: 4

Nº de linhas da matriz: 3

Elementos lidos para o vetor: |1|2|3|4|

Impressão da Matriz:

| 0.5 | 1.0 | 1.5 | 2.0 |

| 1.5 | 3.0 | 4.5 | 6.0 |

| 2.5 | 5.0 | 7.5 | 9.0 |