



Universidade Federal de Pelotas - UFPel
Centro de Desenvolvimento Tecnológico - CDTEC
Curso de Ciência Engenharias da Computação
Algoritmos e Estruturas de Dados I

1- Faça um programa que armazene a informação de várias pessoas.
O programa só deve sair quando o usuário disser que não deseja mais entrar com os dados de outra pessoa.
Antes de sair o programa deve apresentar, de forma organizada, os dados de todas as pessoas. Use a estrutura que segue:

```
typedef struct{  
    char nome[30];  
    int idade;  
    int altura;  
}Pessoa;
```

2- Implementar em C um programa que utilize uma matriz com vetor de ponteiros e que ofereça as seguintes opções para o usuário:

- a) Criar e redimensionar uma matriz $m \times n$, onde n e m são fornecidos pelo usuário;
- b) Realizar a leitura dos elementos da matriz;
- c) Fornecer a soma dos elementos da matriz;
- d) Retornar em um vetor (utilizando ponteiros) os elementos de uma determinada coluna da matriz;
- e) Imprimir a matriz
- f) Sair do programa

Observações:

1.1) A matriz deve ser alocada dinamicamente no programa por meio do uso da função malloc.

1.2) O programa deve ser modularizado e utilizar os seguintes protótipos de subalgoritmos:

- a. `int ** criaMatriz(int m, int n)`
- b. `void leiaMatriz(int **mat, int m, int n)`
- c. `int somaMatriz(int **mat, int m, int n) }`
- d. `int* colunaMatriz(int ** mat, int m, int n, int ncoluna)`
- e. `void liberaMatriz(int **mat, int ncoluna)`
- f. `void imprimeMatriz(int **mat, int m, int n)`
- g. `void imprimeVetor (int *vet, int n) { }`

1.3) O subalgoritmo `int* colunaMatriz(int ** mat, int m, int n, int ncoluna)` deve criar um novo vetor (ponteiro para vetor) e retornar o mesmo para o programa principal que será responsável pela impressão dos valores a partir da chamada de `void imprimeVet (int *vet, int n) { }`.

3- Faça uma agenda capaz de incluir, apagar, buscar e listar quantas pessoas o usuário desejar, porém, toda a informação incluída na agenda deve ficar num único lugar chamado: “`void *pBuffer`”.

Não pergunte para o usuário quantas pessoas ele vai incluir. Não pode alocar espaço para mais pessoas do que o necessário.

Cada pessoa tem nome[10], idade e telefone.

pBuffer

N pessoas	nome1	idade1	Tel1	nome2	idade2	tel2	Nome..n	Idade..n	Tel..n
-----------	-------	--------	------	-------	--------	------	---------	----------	--------