

AVALIAÇÃO 01
BCC32A - Algoritmos e
Estrutura de Dados 1

1. Considerando as informações descritas na página seguinte, escreva as funções abaixo.

a) Pessoa* pessoa_novo(char* nome, int idade, char* email);

Entrada: nome, idade e email da pessoa a ser criada.

Saída: Endereço de memória da pessoa criada dinamicamente contendo as informações passadas por parâmetro.

Processamento: Criar e preencher adequadamente todos os atributos da Pessoa. Não é necessário fazer validações.

b) void pessoa_defineEndereco(Pessoa* p, char* rua, int num, char* estado);

Entrada: endereço de memória da pessoa a ser manipulada, rua, número e estado a serem cadastrados.

Saída: Nenhuma.

Processamento: Cadastrar o endereço recebido pelos parâmetros <rua>, <num> e <estado> na referência de pessoa <p>. Não é necessário fazer validações.

c) void pessoa_cadastrarContato(Pessoa* p, char* celular, char* flagWhats);

Entrada: endereço de memória da pessoa a ser manipulada, celular a ser cadastrado e se o número possui registro no whatsapp.

Saída: Nenhuma.

Processamento: Criar o contato dinamicamente com as informações recebidas e coloca-la na posição adequada do vetor de contatos. Caso o vetor de contatos não tenha sido criado, a função deve criá-lo com um tamanho pré-determinado.

d) void agenda_verificaReduzTamanho(Agenda* agenda);

Entrada: endereço de memória da agenda a ser manipulada.

Saída: sem saída.

Processamento: Verificar se a agenda possui mais do que 50% de espaço ocioso. Caso haja, realocar o vetor com a metade do tamanho atual. Caso a agenda NÃO tenha mais do que 50% de suas posições ociosas, a função não deve fazer nada.

e) int agenda_consultaPorEstado(Agenda* agenda, char* estado, Contato** referencia);

Entrada: endereço de memória da agenda a ser manipulada, string com o estado a ser usado na consulta, **endereço do vetor** que armazenará todos os contatos que moram no estado em questão (Estratégia do scanf).

Saída: Quantidade de registros encontrados e copiados no endereço do vetor recebido por parâmetro.

Processamento: Utilizando o endereço do vetor recebido por parâmetro, copiar todos os contatos que pertençam a pessoa que reside no estado recebido por parâmetro.

Estrutura

```
typedef struct{
    char cep[11];
    int num;
    char estado[3]; // 'PR'
}Endereco;

typedef struct{
    char celular[15];
    char whatsapp[2]; // 'S' ou 'N'
} Contato;


```

```
typedef struct{
    char* nome;
    int idade;
    char* email;
    Endereco endereco;
    Contato** contatos;
    int contatoTam; //tamanho do vetor contatos
    int contatoQtde; //tamanho do vetor contatos
}Pessoa;

typedef struct{
    char descricao[20]; // 'Pessoal', 'Profissional'
    Pessoa** pessoas; //Vetor(*) de ponteiros de Pessoa (Pessoa*)
    int pessoasTam; //Tamanho do vetor pessoas
    int pessoasQtde; //Qtde de pessoas no vetor
}Agenda;
```

Representação

