

Algoritmos e Estruturas de Dados I

Prof. Lúcio Mauro Pereira

06/11/2024

Aulas anteriores

Construindo um arquivo de cabeçalho (biblioteca)

```
#ifndef DATA_H_INCLUDED
```

```
#define DATA_H_INCLUDED
```

```
// código do cabeçalho
```

```
#endif // DATA_H_INCLUDED
```

Construindo o TAD Data

```
#ifndef DATA_H_INCLUDED
#define DATA_H_INCLUDED

typedef struct {
    int dia;
    int mes;
    int ano;
} Data;

#endif//DATA_H_INCLUDED
```

Construindo o TAD Data

```
#ifndef DATA_H_INCLUDED
#define DATA_H_INCLUDED

typedef struct {
    int dia;
    int mes;
    int ano;
} Data;

int diasMes(Data data) {
    int dias[]={31,28,31,30,31,30,31,31,30,31,30,31};
    return dias[ data.mes-1 ];
}

char* mesExtenso(Data data) {
    char* mes={"janeiro","fevereiro","março","abril","maio",
"junho","julho","agosto","setembro","outubro","novembro","dezembro"};
    return mes[ data.mes-1 ];
}

void escreveData(Data data){
    printf("%d/%d/%d",
        data.dia,
        data.mes,
        data.ano);
}

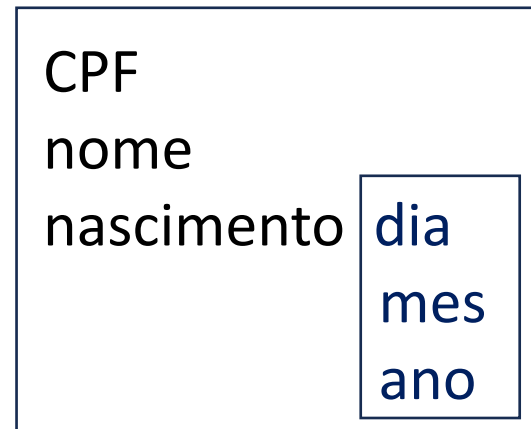
void leiaData(Data* data){
    printf("\ndd/mm/aaaa: ");
    scanf("%d/%d/%d",
        &data->dia,
        &data->mes,
        &data->ano);
}

#endif//DATA_H_INCLUDED
```

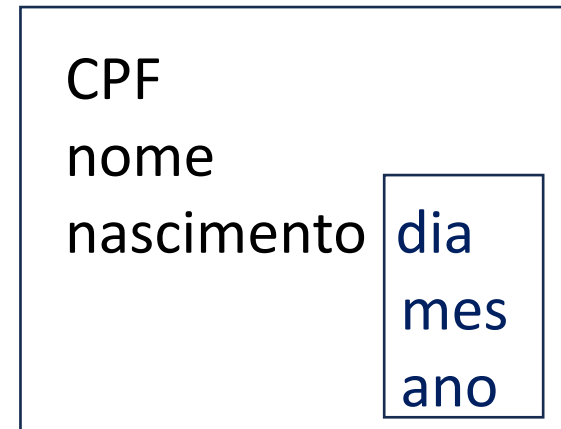

Construindo o TAD Pessoa

```
#include "data.h"
```

```
typedef struct  
{  
    char CPF[14];  
    char nome[50];  
    Data nascimento;  
} Pessoa;
```



0



1

```

#ifndef PESSOA_H_INCLUDED
#define PESSOA_H_INCLUDED
#include "data.h"
typedef struct {
    char nome[50];
    Data nascimento;
} Pessoa;

int TAM = 0;

int tamanho(char* arq) {
    FILE* arqTamanho = fopen(arq, "rb");
    if(arqTamanho == NULL){
        arqTamanho = fopen(arq, "wb");
        TAM=0;
        fprintf(arqTamanho, "%i", TAM);
    } else {
        fscanf(arqTamanho, "%i", &TAM);
    }
    fclose(arqTamanho);
    return TAM;
}

void escrevaPessoa(Pessoa PESSOAS[], int i){
    escrevaEstPessoa( PESSOAS[i]);
}

```

```

void escrevaEstPessoa(Pessoa PESSOA){
    printf("\nNome: %s", PESSOA.nome );
    printf("\nData de Nascimento: %d/%d/%d ",
        PESSOA.Nascimento.dia,
        PESSOA.Nascimento.mes,
        PESSOA.Nascimento.ano );
}

void abertura(Pessoa PESSOAS[]){
    printf("\nControle de Pessoas\n");
    TAM = tamanho("tamanhoArq.dat");
    carregaPessoas(PESSOAS);
}

void cadastrePessoa(Pessoa PESSOAS[]){
    fflush(stdin); //Linux __fpurge(stdin);
    printf("\nNome: ");
    fgets(PESSOAS[TAM].nome, MAX_STR, stdin);
    printf("\nData de Nascimento [dd/mm/aaaa]: ");
    scanf("%d/%d/%d",
        &PESSOAS[TAM].Nascimento.dia,
        &PESSOAS[TAM].Nascimento.mes,
        &PESSOAS[TAM].Nascimento.ano );
    TAM++;
}

#endif // PESSOA_H_INCLUDED

```

```
//Ao abrir – Por exemplo, em abertura()
void carregaPessoas(Pessoa PESSOAS[]) {
    FILE* arqPessoas = fopen("pessoas.dat", "rb+");

    if(arqPessoas == NULL){
        arqPessoas = fopen("pessoas.dat","wb+");
    }

    fread(PESSOAS, sizeof(Pessoa),TAM,arqPessoas);

    fclose(arqPessoas);
}
```

```
//Ao fechar – Por exemplo, em despedida()
void gravaPessoas(Pessoa PESSOAS[]){
    FILE* arqPessoas = fopen("pessoas.dat", "wb");
    fwrite(PESSOAS, sizeof(Pessoa), TAM, arqPessoas);
    fclose(arqPessoas);

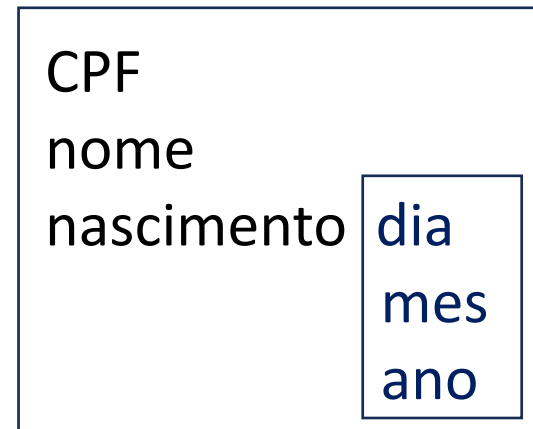
    FILE* arqTamanho = fopen("tamanhoArq.dat", "wb");
    fprintf(arqTamanho, "%i", TAM);
    fclose(arqTamanho);
}
```


Questões

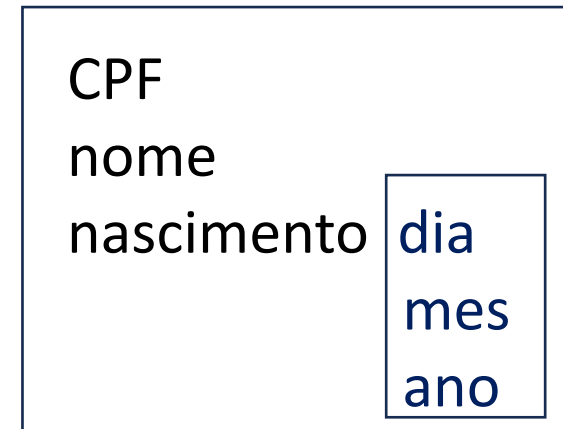
Chave de Pesquisa: nome

```
void pesquisaPessoaNome( Pessoa PESSOAS[], char* nome )  
{
```

```
}
```



0



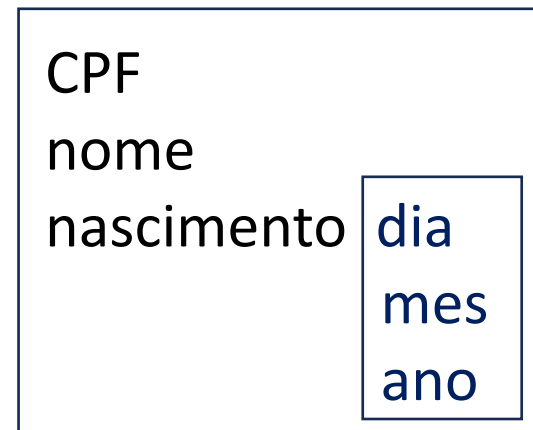
1

Chave de Pesquisa: CPF

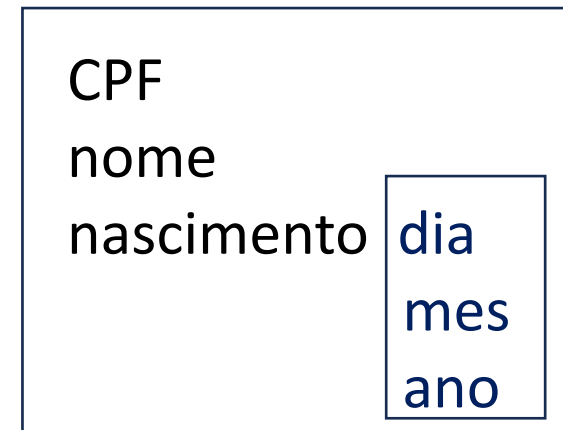
Retorna a posição da estrutura

```
Pessoa* pesquisaPessoaCPF( Pessoa PESSOAS[], char* cpf )  
{
```

```
}
```



0

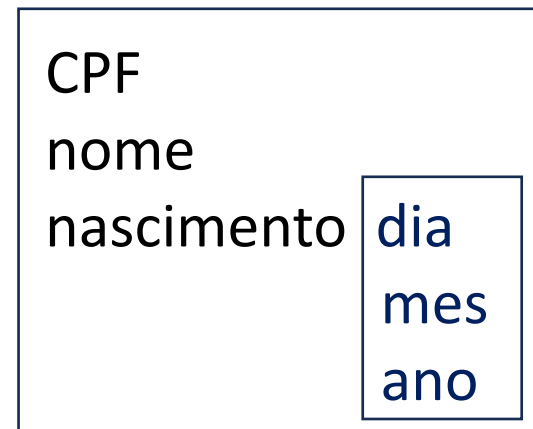


1

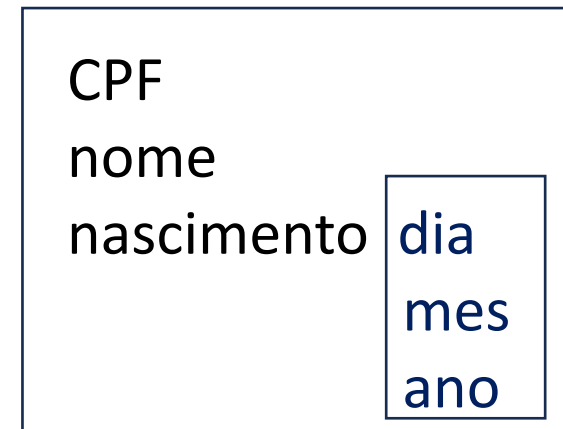
Chave de Pesquisa: nome
Retorna a posição do arranjo

```
int pesquisaPessoaCPF( Pessoa PESSOAS[], char* cpf )  
{
```

```
}
```



0

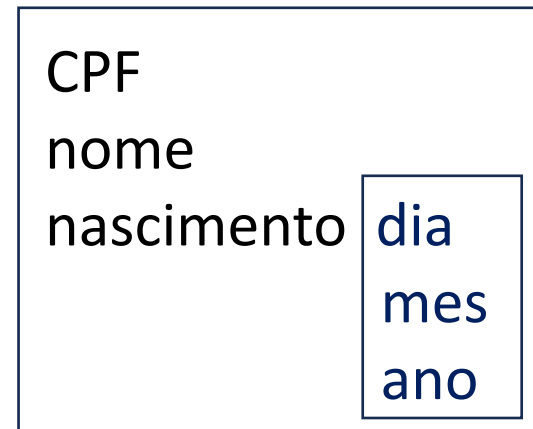


1

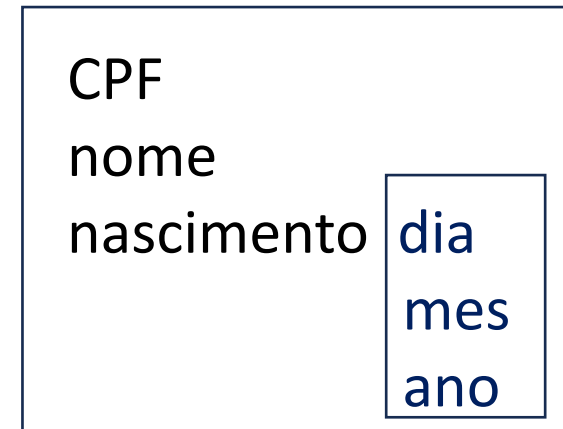
Exclusão. Chave: CPF
shift -1: desloca os demais à esquerda

```
bool deletaPessoa ( Pessoa PESSOAS[], char* CPF )  
{
```

```
}
```



0



1