

Algoritmos e Estruturas de Dados I

Prof. Lúcio Mauro Pereira

31/out e 1º/nov /2024

Tipo Definido pelo Usuário
Tipo Abstrato de Dados

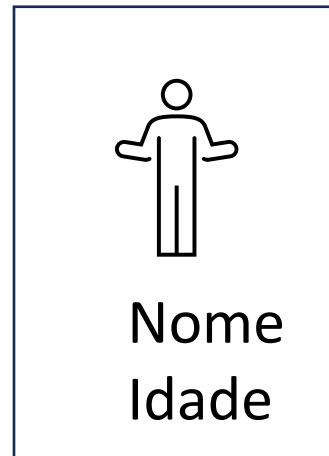
Implemente

Estruturas: Abstração de Dados

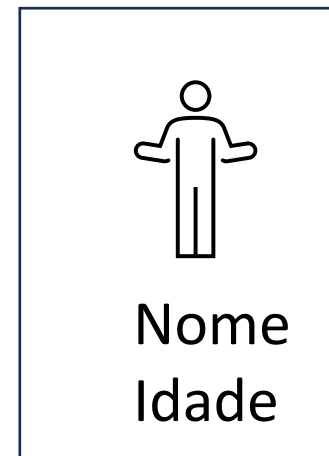
```
#define MAX_STR 50
```

```
#define MAX 100
```

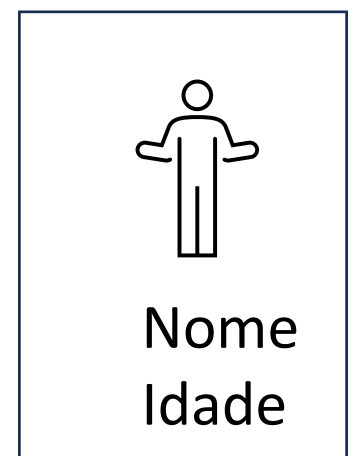
```
int TAM = 0;
```



0



1

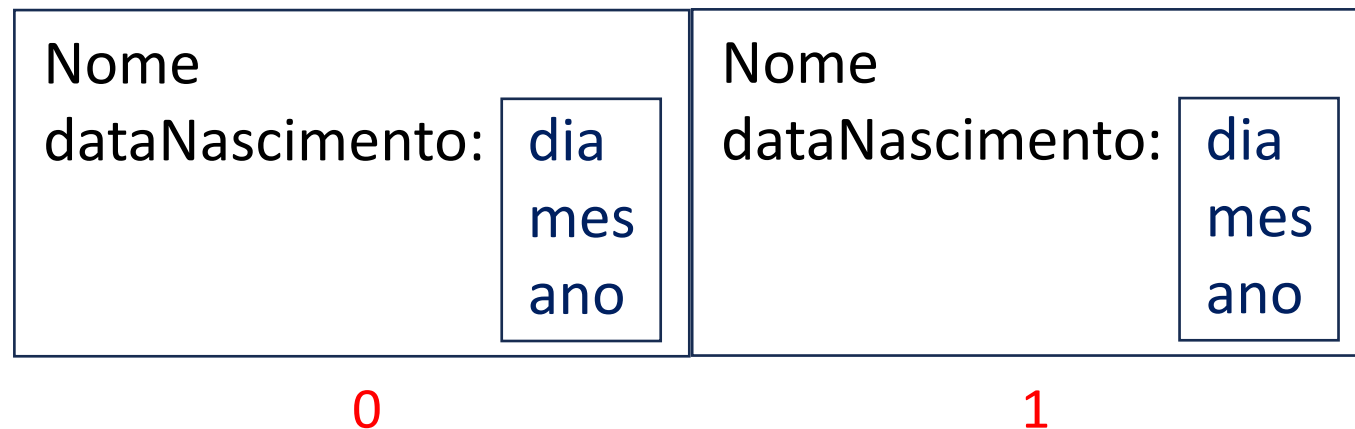


2

Estruturas: Abstração de Dados

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

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



Estruturas: Abstração de Dados

```
#define MAX_STR 50
```

```
#define MAX 100
```

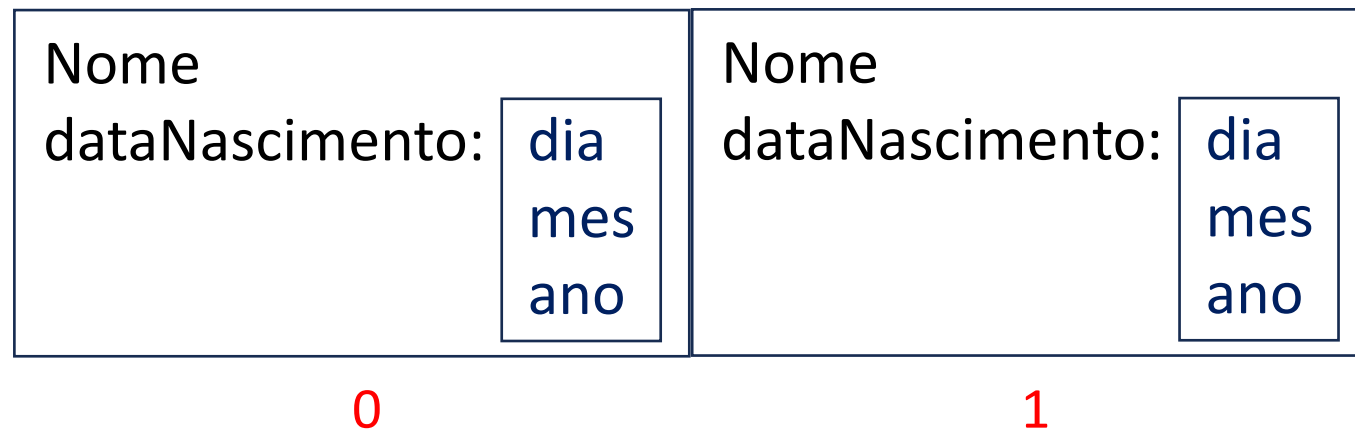
```
int TAM = 0;
```

```
typedef struct
```

```
{  
    int dia;  
    int mes;  
    int ano;  
} Data;
```

```
typedef struct
```

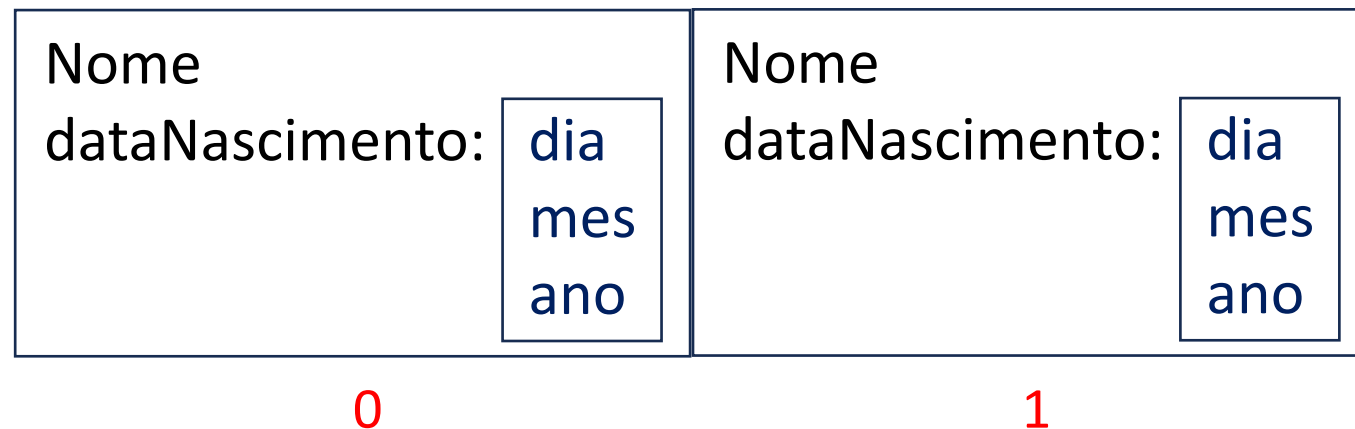
```
{  
    char nome[MAX_STR];  
    Data dataNascimento;  
} Pessoa;
```



Escrevendo Pessoas

```
void escrevaPessoa(Pessoa PESSOAS[], int i){  
    printf("\nNome: %s", PESSOAS[i].nome );  
    printf("\nData de Nascimento: %d/%d/%d ",  
  
    );  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```



Escrevendo Pessoas

```
void escrevaPessoa(Pessoa PESSOAS[], int i){  
    printf("\nNome: %s", PESSOAS[i].nome );  
    printf("\nData de Nascimento: %d/%d/%d ",  
        PESSOAS[i].dataNascimento.dia,  
        PESSOAS[i].dataNascimento.mes,  
        PESSOAS[i].dataNascimento.ano  
    );  
}
```

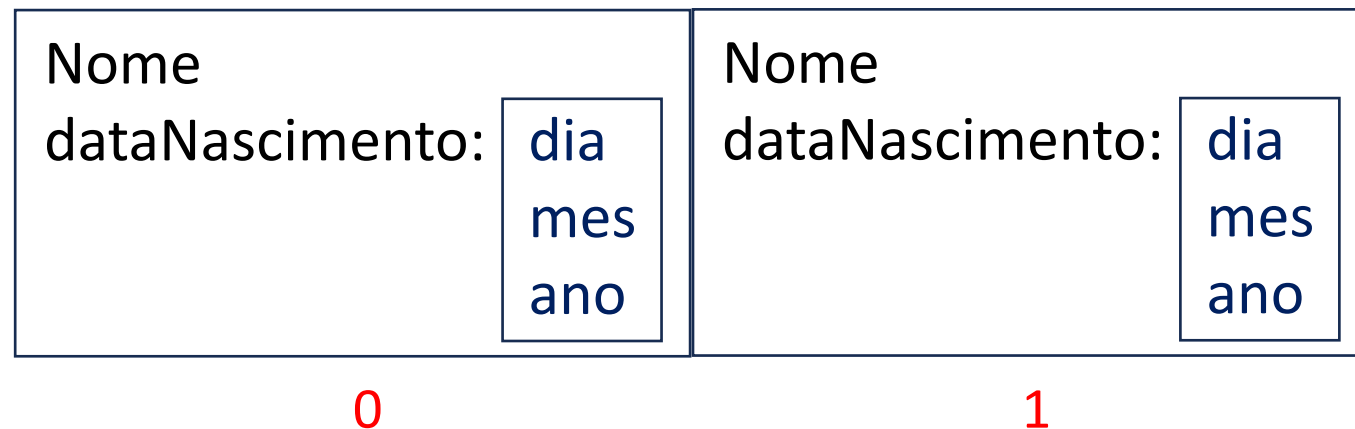
```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

<div>Nome dataNascimento: <div>dia mes ano</div></div>	<div>Nome dataNascimento: <div>dia mes ano</div></div>
0	1

Escrevendo Pessoas

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

```
#define MAX_STR 50  
#define MAX     100  
int     TAM =    0;
```

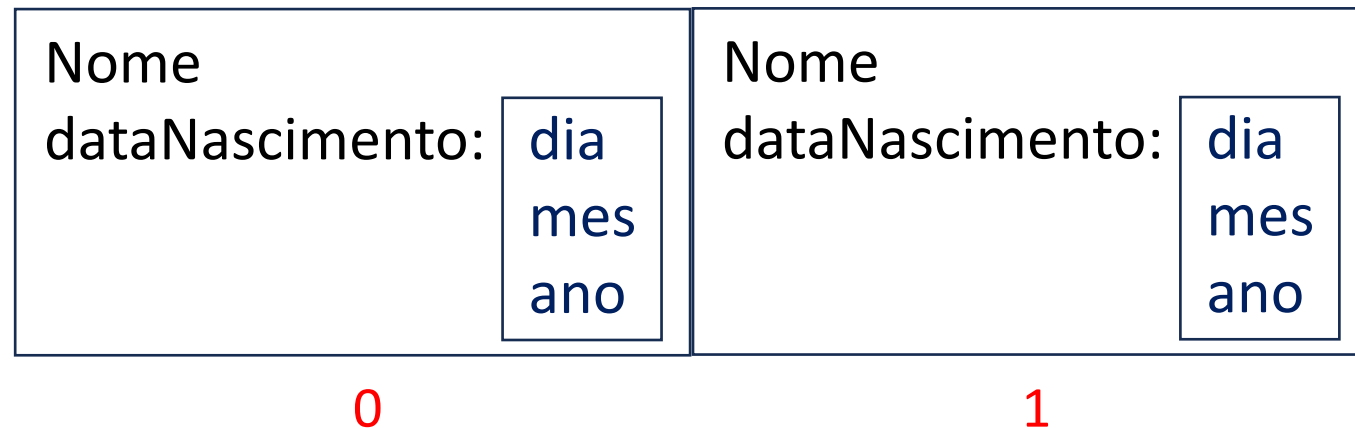


Escrevendo Pessoas

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

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

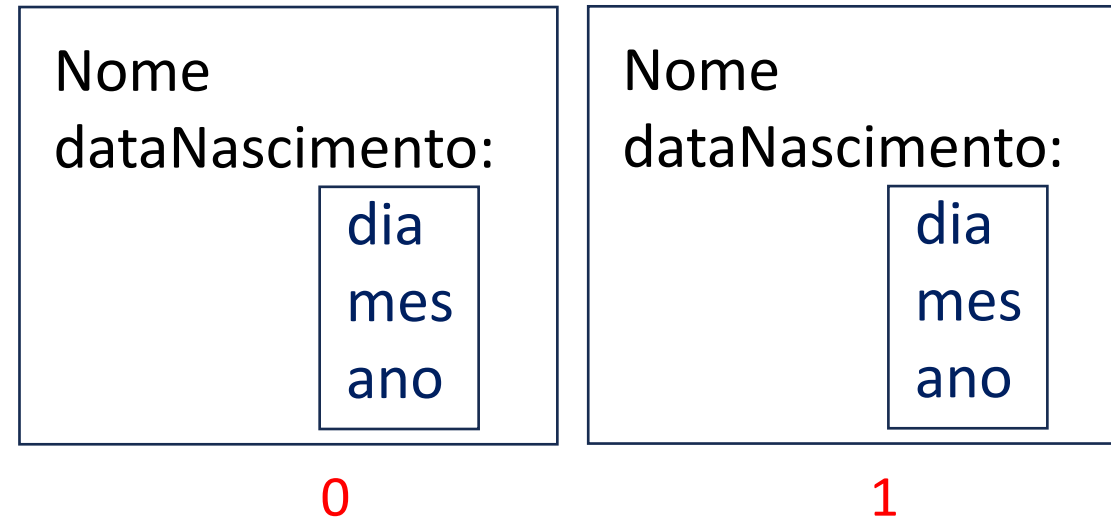
```
void escrevaEstPessoa(Pessoa PESSOA){  
    printf("\nNome: %s", PESSOA.nome );  
    printf("\nData de Nascimento: %d/%d/%d ",  
        PESSOA.dataNascimento.dia,  
        PESSOA.dataNascimento.mes,  
        PESSOA.dataNascimento.ano  
    );  
}
```



Lendo Pessoas

```
void cadastrePessoa(Pessoa PESSOAS[]){  
  
    fflush(stdin); //Linux __fpurge(stdin);  
    printf("\nNome: " );  
    fgets(PESSOAS[TAM].nome , MAX_STR, stdin);  
  
    printf("\nData de Nascimento: ");  
  
    TAM++;  
}
```

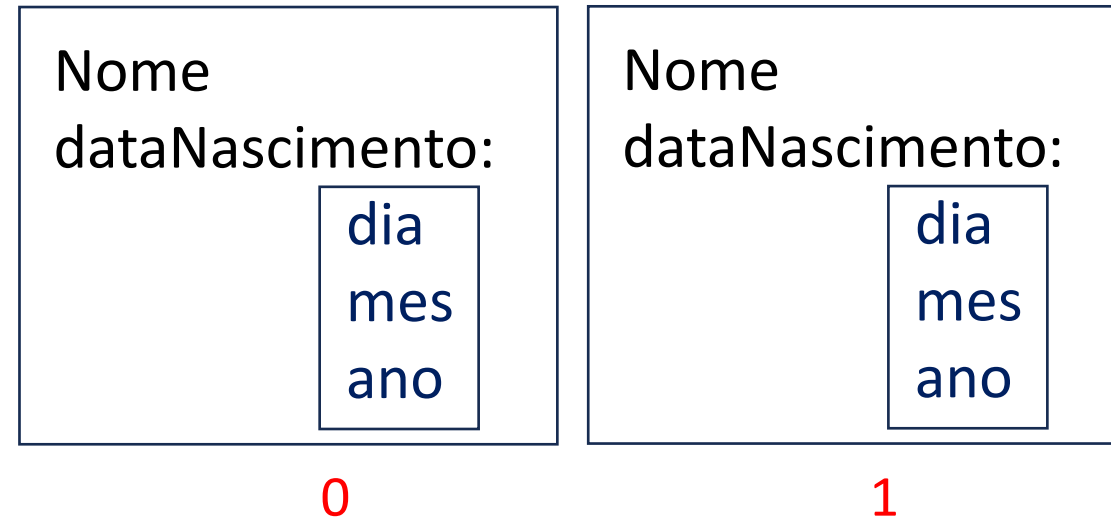
```
#define MAX_STR 50  
#define MAX      100  
int TAM = 0;
```



Lendo Pessoas

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

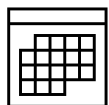
```
#define MAX_STR 50  
#define MAX      100  
int TAM = 0;
```



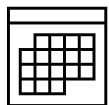
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

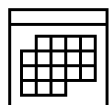
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



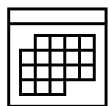
Dia
Mês
Ano

integer **diasMes()**
string **mesExtenso()**
boolean **ehBissexto()**
boolean **dataValida()**
string **diaSemana()**
...

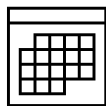
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

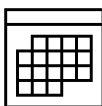
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



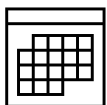
Dia
Mês
Ano

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

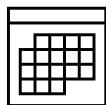
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

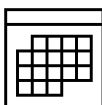
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



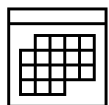
Dia
Mês
Ano

integer diasMes()
string mesExtenso()
boolean ehBissexto()
boolean dataValida()
string diaSemana()
...

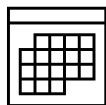
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

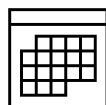
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



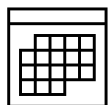
Dia
Mês
Ano

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

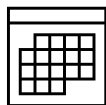

Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

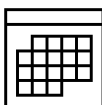
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano

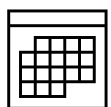


Dia
Mês
Ano

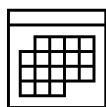
integer diasMes()
string mesExtenso()
boolean ehBissexto()
boolean dataValida()
string diaSemana()
...

Modelando um Tipo Abstrato de Dados: **Data**

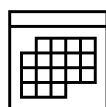
Abstração de **Dados** e Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



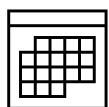
Dia
Mês
Ano

```
bool ehBissexto(Data data)
{
    return ;
}
```

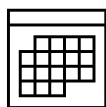
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

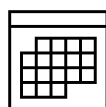
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano

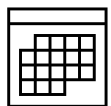


Dia
Mês
Ano

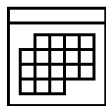
integer diasMes()
string mesExtenso()
boolean ehBissexto()
boolean dataValida()
string diaSemana()
...

Modelando um Tipo Abstrato de Dados: **Data**

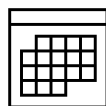
Abstração de **Dados** e Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



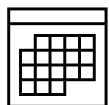
Dia
Mês
Ano

```
bool dataValida(Data data)
{
    return ;
}
```

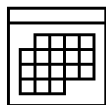
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

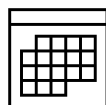
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



Dia
Mês
Ano

integer diasMes()
string mesExtenso()
boolean ehBissexto()
boolean dataValida()
string diaSemana()

...

Testando

```
int main(){
    Data hoje;
```

```
#define MAX_STR 50
#define MAX 100
int TAM = 0;
```

```
return 0;
```

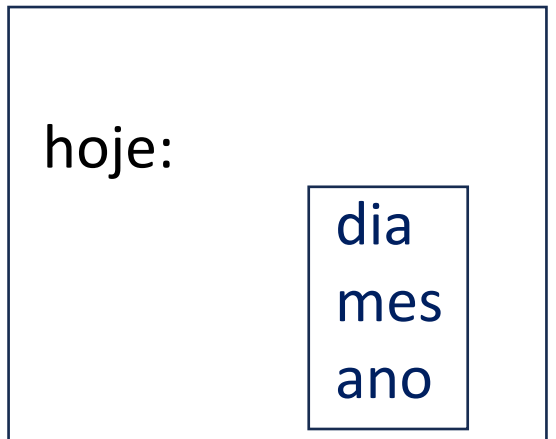
hoje:

dia
mes
ano

Testando

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```



Testando

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );
```

```
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

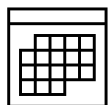
hoje:

dia
mes
ano

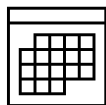
Modelando um Tipo Abstrato de Dados: **Data**

Abstração de **Dados**

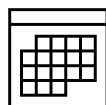
Abstração de **Operações**



Dia
Mês
Ano



Dia
Mês
Ano



Dia
Mês
Ano

integer	diasMes()
string	mesExtenso()
boolean	ehBissexto()
boolean	dataValida()
void	escreveData()
...	

Testando

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

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );
```

```
    return 0;
```

```
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

Testando

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

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

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

```
int main(){  
  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

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

```
void leiaData(Data* data){  
  
  
}
```

```
int main(){  
  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

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

```
void leiaData(Data* data){  
    printf("\ndd/mm/aaaa: ");  
}
```

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

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

```
void leiaData(Data* data){  
    printf("\ndd/mm/aaaa: ");  
    scanf("%d/%d/%d",  
  
}
```

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

```
void escreveData(Data data){  
    printf("\n%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);  
}
```

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

```
void escreveData(Data data){  
    printf("\n%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);  
}
```

```
int main(){  
    Data hoje;  
    hoje.dia = 1;  
    hoje.mes = 11;  
    hoje.ano = 2024;  
    escreveData( hoje );  
  
    Data nascimento;  
    leiaData(&nascimento);  
    escreveData( nascimento );  
  
    return 0;  
}
```

```
#define MAX_STR 50  
#define MAX 100  
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

```
void escreveData(Data data){
    printf("\n%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);
}

int main(){
    Data hoje;
    hoje.dia = 1;
    hoje.mes = 11;
    hoje.ano = 2024;
    escreveData( hoje );

    Data nascimento;
    leiaData(&nascimento);
    escreveData( nascimento );
    printf("\nO mes %s tem %i dias",

    return 0;
}
```

```
#define MAX_STR 50
#define MAX 100
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

Testando

```
void escreveData(Data data){
    printf("\n%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);
}

int main(){
    Data hoje;
    hoje.dia = 1;
    hoje.mes = 11;
    hoje.ano = 2024;
    escreveData( hoje );

    Data nascimento;
    leiaData(&nascimento);
    escreveData( nascimento );
    printf("\nO mes %s tem %i dias",
        mesExtenso(nascimento),
        diasMes(nascimento));
    return 0;
}
```

```
#define MAX_STR 50
#define MAX 100
int TAM = 0;
```

hoje:

dia
mes
ano

nascimento:

dia
mes
ano

O tipo Data

```
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 ];
```

```
}
```

O tipo Data

```
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("\n%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);
}
```


O tipo Data

Abstração de dados:

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

O tipo Data

Abstração de operações:

```
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("\n%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);  
}
```

Implementando um Tipo Abstrato de Dados

O arquivo de cabeçalho .h

Selezione (em Code::blocks)

File

New

File

C/C++ header

data.h

Construa 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 ];
}

#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 ];
}

#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("\n%d/%d/%d",
        data.dia,
        data.mes,
        data.ano);
}
```

```
#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("\n%d/%d/%d",
        data.dia,
        data.mes,
        data.ano);
}

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

#endif//DATA_H_INCLUDED
```

Utilizando o TAD

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <locale.h>

#include "obj\\data.h"

#define MAX_STR 50
#define MAX 100
int TAM = 0;
```

Utilizando o TAD

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <locale.h>

#include "obj\\data.h"

#define MAX_STR 50
#define MAX 100
int TAM = 0;
```

```
int main()
{
    Data nascimento;

    leiaData(&nascimento);
    escreveData( nascimento );

    printf("\nO mes %s tem %i dias",
           mesExtenso(nascimento),
           diasMes(nascimento));

    return 0;
} // fim main()
```

Tente:

Acrescente ao TAD Data as seguintes operações:

Verificar se uma data é válida ou não

Verificar se a data corresponde a um ano bissexto ou não

```
int main() {  
    Data nascimento;  
    leiaData(&nascimento);  
    escreveData( nascimento );  
    if( dataValida(nascimento) ) {  
        printf("\nO mes %s tem %i dias", mesExtenso(nascimento), diasMes(nascimento) );  
        if( ehBissexto(nascimento) ) printf("\nAno bissexto");  
        else printf("\nAno não bissexto");  
    }  
    return 0;  
} // fim main()
```

Trabalho Prático Final:

Implemente em arquivo de cabeçalho:

Tipo Data (data.h)

Tipo Pessoa (pessoa.h)