

Fundamentos de Programação 1

Linguagem C

**“ Struct,
Union, Enum, Typedef”**

Slides 11

Prof. SIMÃO

Exemplo em C – Versão 1

```
#include <stdio.h>
```

```
struct Pessoa
```

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

```
int Calc_Idade ( struct Pessoa p, int a )
```

```
{  
    int idad;  
  
    idad = a - p.ano;  
  
    return idad;  
}
```

```
int main()
```

```
{  
    struct Pessoa Einstein, Newton;  
  
    Einstein.dia  = 14;  
    Einstein.mes  = 3;  
    Einstein.ano  = 1879;  
  
    Newton.dia    = 4;  
    Newton.mes    = 1;  
    Newton.ano    = 1643;  
  
    Einstein.idade = Calc_Idade ( Einstein, 2007 );  
    Newton.idade  = Calc_Idade ( Newton, 2007 );  
  
    printf ( "A idade de Einstein seria %d \n", Einstein.idade );  
    printf ( "A idade de Newton seria %d \n", Newton.idade );  
  
    getchar();  
    return 0;  
}
```

Exemplo de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

```
int Calc_Idade (struct Pessoa p, int d, int m, int a)  
{  
    int idad = a - p.ano;  
  
    if ( p.mes > m )  
    {  
        idad = idad - 1;  
    }  
    else  
    {  
        if ( p.mes == m )  
        {  
            if ( p.dia > d )  
            {  
                idad = idad - 1;  
            }  
        }  
    }  
    return idad;  
}
```

```
int main()  
{  
    struct Pessoa Einstein, Newton;  
  
    Einstein.dia = 14;  
    Einstein.mes = 3;  
    Einstein.ano = 1879;  
  
    Newton.dia = 4;  
    Newton.mes = 1;  
    Newton.ano = 1643;  
  
    Einstein.idade = Calc_Idade (Einstein, 27, 9, 2007);  
    Newton.idade = Calc_Idade (Newton, 27, 9, 2007);  
  
    printf("A idade de Einstein seria %d \n", Einstein.idade);  
    printf("A idade de Newton seria %d \n", Newton.idade);  
  
    getchar();  
    return 0;  
}
```

Exemplo de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

```
int Calc_Idade (struct Pessoa p, int dia, int mes, int ano)  
{  
    int idade = ano - p.ano;  
  
    if ( p.mes > mes )  
    {  
        idade = idade - 1;  
    }  
    else  
    {  
        if ( p.mes == mes )  
        {  
            if ( p.dia > dia )  
            {  
                idade = idade - 1;  
            }  
        }  
    }  
    return idade;  
}
```

```
int main()  
{  
    struct Pessoa Einstein, Newton;  
  
    Einstein.dia = 14;  
    Einstein.mes = 3;  
    Einstein.ano = 1879;  
  
    Newton.dia = 4;  
    Newton.mes = 1;  
    Newton.ano = 1643;  
  
    Einstein.idade = Calc_Idade (Einstein, 27, 9, 2007);  
    Newton.idade = Calc_Idade (Newton, 27, 9, 2007);  
  
    printf("A idade de Einstein seria %d \n", Einstein.idade);  
    printf("A idade de Newton seria %d \n", Newton.idade);  
  
    getchar();  
    return 0;  
}
```

Exemplo de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

```
int Calc_Idade (struct Pessoa p, int dia, int mes, int ano)  
{  
    p.idade = ano - p.ano;  
  
    if ( p.mes > mes )  
    {  
        p.idade = p.idade - 1;  
    }  
    else  
    {  
        if ( p.mes == mes )  
        {  
            if ( p.dia > dia )  
            {  
                p.idade = p.idade - 1;  
            }  
        }  
    }  
    return p.idade;  
}
```

```
int main()  
{  
    struct Pessoa Einstein, Newton;  
  
    Einstein.dia = 14;  
    Einstein.mes = 3;  
    Einstein.ano = 1879;  
  
    Newton.dia = 4;  
    Newton.mes = 1;  
    Newton.ano = 1643;  
  
    Einstein.idade = Calc_Idade (Einstein, 27, 9, 2007);  
    Newton.idade = Calc_Idade (Newton, 27, 9, 2007);  
  
    printf("A idade de Einstein seria %d \n", Einstein.idade);  
    printf("A idade de Newton seria %d \n", Newton.idade);  
  
    getchar();  
    return 0;  
}
```

Exemplo de *Struct*

```
#include <stdio.h>
struct Pessoa
{
    char nome[30];
    int dia;
    int mes;
    int ano;
    int idade;
};
```

```
int Calc_Idade (struct Pessoa p, int dia, int mes, int ano)
{
    p.idade = ano - p.ano;

    if ( p.mes > mes )
    {
        p.idade = p.idade - 1;
    }
    else
    {
        if ( p.mes == mes )
        {
            if ( p.dia > dia )
            {
                p.idade = p.idade - 1;
            }
        }
    }
    return p.idade;
}
```

```
int main()
{
    struct Pessoa Einstein, Newton;

    // Einstein.nome = "Albert Einstein";
    strcpy ( Einstein.nome, "Albert Einstein" );

    Einstein.dia = 14;
    Einstein.mes = 3;
    Einstein.ano = 1879;

    // Newton.nome = "Isaac Newton";
    strcpy ( Newton.nome, "Isaac Newton" );

    Newton.dia = 4;
    Newton.mes = 1;
    Newton.ano = 1643;

    Einstein.idade = Calc_Idade ( Einstein, 27, 9, 2007);
    Newton.idade = Calc_Idade ( Newton, 27, 9, 2007);

    printf("A idade de %s seria %d \n", Einstein.nome, Einstein.idade );
    printf("A idade de %s seria %d \n", Newton.nome, Newton.idade );

    getchar();
    return 0;
}
```

Exemplo 2 de Struct

```
#include <stdio.h>
#include <string.h>
```

```
struct Pessoa
{
    char nome[30];
    int dia;
    int mes;
    int ano;
    int idade;
};
```

p é um parâmetro por valor aqui...

```
int Calc_Idade (struct Pessoa p, int dia, int mes, int ano)
{
    int idade = ano - p.ano;
    if ( p.mes > mes )
    {
        idade = idade - 1;
    }
    else
    {
        if (p.mes == mes)
        {
            if (p.dia > dia)
            {
                idade = idade - 1;
            }
        }
    }
    return idade;
}
```

```
int main()
{
    struct Pessoa Lista_Genios[2]

    strcpy ( Lista_Genios[0].nome, "Albert Einstein" );
    Lista_Genios[0].dia = 14;
    Lista_Genios[0].mes = 3;
    Lista_Genios[0].ano = 1879;

    strcpy ( Lista_Genios[1].nome, "Isaac Newton" );
    Lista_Genios[1].dia = 4;
    Lista_Genios[1].mes = 1;
    Lista_Genios[1].ano = 1643;

    Lista_Genios[0].idade = Calc_Idade (Lista_Genios[0], 8, 2, 2007);
    Lista_Genios[1].idade = Calc_Idade (Lista_Genios[1], 8, 2, 2007);

    printf("A idade de %s seria %d \n", Lista_Genios[0].nome
        Lista_Genios[0].idade );

    printf("A idade de %s seria %d \n", Lista_Genios[1].nome
        Lista_Genios[1].idade );

    getchar();
    return 0;
}
```

Exemplo 3 de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    char nome[30];  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

p é um **parâmetro** por referência aqui...

```
void Calc_Idade (struct Pessoa *p, int dia, int mes, int ano)  
{  
    (*p).idade = ano - (*p).ano;  
  
    if ( (*p).mes > mes )  
    {  
        (*p).idade = (*p).idade - 1;  
    }  
    else  
    {  
        if ( (*p).mes == mes )  
        {  
            if ( (*p).dia > dia )  
            {  
                (*p).idade = (*p).idade - 1;  
            }  
        }  
    }  
}
```

```
int main()
```

```
{  
    struct Pessoa Einstein, Newton;  
  
    strcpy ( Einstein.nome, "Albert Einstein" );  
    Einstein.dia = 14;  
    Einstein.mes = 3;  
    Einstein.ano = 1879;  
  
    strcpy ( Newton.nome, "Newton");  
    Newton.dia = 4;  
    Newton.mes = 1;  
    Newton.ano = 1643;  
  
    Calc_Idade ( &Einstein, 28, 9, 2007 );  
    Calc_Idade ( &Newton, 28, 9, 2007 );  
  
    printf("A idade de %s seria %d \n", Einstein.nome  
        Einstein.idade );  
    printf("A idade de %s seria %d \n", Newton.nome  
        Newton.idade );  
  
    getchar();  
    return 0;  
}
```

Usa-se a forma *(*p).variável*

Exemplo 4 de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    char nome[30];  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

p é um parâmetro por referência aqui...

```
void Calc_Idade (struct Pessoa *p, int dia, int mes, int ano)  
{  
    p->idade = ano - p->ano;  
  
    if ( p->mes > mes )  
    {  
        p->idade = p->idade - 1;  
    }  
    else  
    {  
        if ( p->mes == mes )  
        { if ( p->dia > dia )  
            {  
                p->idade = p->idade - 1;  
            }  
        }  
    }  
}
```

```
int main()
```

```
{  
    struct Pessoa Einstein, Newton;  
  
    strcpy ( Einstein.nome, "Albert Einstein" );  
    Einstein.dia = 14;  
    Einstein.mes = 3;  
    Einstein.ano = 1879;  
  
    strcpy ( Newton.nome, "Newton");  
    Newton.dia = 4;  
    Newton.mes = 1;  
    Newton.ano = 1643;  
  
    Calc_Idade ( &Einstein, 28, 9, 2007 );  
    Calc_Idade ( &Newton, 28, 9, 2007 );  
  
    printf ("A idade de %s seria %d \n", Einstein.nome  
           Einstein.idade );  
    printf ("A idade de %s seria %d \n", Newton.nome  
           Newton.idade );  
  
    getchar();  
    return 0;  
}
```

Em vez da forma *(*p).variável*,
usa-se mais a forma *p->variável*.

Exemplo 5 de Struct

```
#include <stdio.h>
```

```
struct Pessoa  
{  
    char nome[30];  
    int dia;  
    int mes;  
    int ano;  
    int idade;  
};
```

p é um parâmetro por referência aqui...

```
void Calc_Idade (struct Pessoa *p, int dia, int mes, int ano)  
{  
    p->idade = ano - p->ano;  
  
    if ( p->mes > mes )  
    {  
        p->idade = p->idade - 1;  
    }  
    else  
    {  
        if ( p->mes == mes )  
        {  
            if ( p->dia > dia )  
            {  
                p->idade = p->idade - 1;  
            }  
        }  
    }  
}
```

```
int main()  
{  
    struct Pessoa Lista_Genios[2];  
  
    strcpy ( Lista_Genios[0].nome, "Einstein");  
    Lista_Genios[0].dia = 14;  
    Lista_Genios[0].mes = 3;  
    Lista_Genios[0].ano = 1879;  
  
    strcpy ( Lista_Genios[1].nome, "Newton");  
    Lista_Genios[1].dia = 4;  
    Lista_Genios[1].mes = 1;  
    Lista_Genios[1].ano = 1643;  
  
    Calc_Idade ( &Lista_Genios[0], 28, 9, 2007);  
    Calc_Idade ( &Lista_Genios[1], 28, 9, 2007);  
  
    printf("A idade de %s seria %d \n", Lista_Genios[0].nome,  
        Lista_Genios[0].idade );  
  
    printf("A idade de %s seria %d \n", Lista_Genios[1].nome,  
        Lista_Genios[1].idade );  
  
    getchar();  
    return 0;  
}
```

Outro exemplo

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

struct Pessoa
{
    int idade;
    char nome [ 100 ];
    char sexo;
};

int quantidmaior (struct Pessoa fs[], int tam, float med)
{
    int quant = 0;

    int cont = 0;

    for ( cont = 0; cont < tam; cont = cont + 1 )
    {
        if ( fs[cont].idade > med)
        {
            quant = quant + 1;
        }
    }

    return quant;
}
```

```
int main()
{
    struct Pessoa Funcs[3];

    int quant, contaid = 0;
    float media;

    printf ( "Programa dos dados de funcionários. \n \n \n" );

    int cont;
    for ( cont = 0; cont < 3; cont = cont + 1 )
    {
        printf ("Digite o nome do funcionario número %i: \n", cont+1 );
        fflush ( stdin );
        gets ( Funcs[cont].nome );

        printf ( "Digite a idade do funcionario número %i: \n", cont+1);
        fflush ( stdin );
        scanf ( "%i", &Funcs[cont].idade );

        printf ( "Digite o sexo (f/m) do funcionario número %i: \n", cont+1 );
        fflush ( stdin );
        scanf ( "%c", &Funcs[cont].sexo);
        contaid = contaid + Funcs[cont].idade;

        printf ("\n");
    }

    media = (float) contaid/cont;
    // afinal, cont vale 3

    quant = quantidmaior (Funcs, cont, media); // afinal, cont vale 3

    printf ( " A média de idade é %.2f: \n", media );
    printf ( " A quantidade de funcionário com idade acima da média é %i: \n", quant );

    printf ("\n \n");
    system ("Pause");
    return 0;
}
```

Exercícios

- Fazer um programa que receba três nomes de no máximo 15 caracteres cada um (nomes com mais de 15 caracteres devem ser rejeitados) e as idades da respectivas pessoas em um vetor de estruturas de dados. Após o recebimento, listar os 3 nomes e idades que nela foram armazenados.
- Fazer um programa de 'diálogo de login' onde seja possível cadastrar no máximo 10 nomes de usuário e suas respectivas senhas (nomes de usuário repetidos devem ser descartados). No diálogo de login, o programa deve testar se o usuário fornecido existe e se a sua senha confere.

Fazer exercícios:

- Como os proposto em (Forbellone & Eberspächer) sobre *estruturas*. Entretanto, além dos algoritmos, fazer também (ou tão somente, caso queira) os respectivos programas.

(Forbellone & Eberspächer)

André Luiz Villar Forbellone & Henri Frederico Eberspächer. *Lógica de Programação : A construção de Algoritmos e Estruturas de Dados*. Makron Books. 1993

Estudar a teoria/exemplos e fazer exercícios sobre:

- Enumerações em C (*enum*)
- Uniões em C (*union*)
- Definição de tipo (*typedef*)

A quem interessar possa.

- Visitar o seguinte sítio:

<http://www.dainf.cefetpr.br/~robson/>

- Ir em:
 - Conteúdo.
 - Profissional.
 - Disciplinas DAINF
 - » Computação I.