

Estruturas e funções

Murilo Dantas

Estruturas e funções

- É possível ter estruturas como parâmetro
- É possível retornar estruturas
- Exemplo 8
 - ▶ Fazer uma função chamada maiorIdade que receba uma struct Pessoa e retorne 0 caso a pessoa seja menor que 21 anos e 1 caso contrário.

Exemplo 8

```
#include <iostream>

using namespace std;

struct Pessoa {
    string nome;
    int idade;
};

int maiorIdade(struct Pessoa p)
{
    if (p.idade < 21)
        return 0;
    return 1;
}

int main()
{
    struct Pessoa alguem = {"Fulano", 28};

    if(maiorIdade(alguem))
        cout << "Pessoa maior de idade\n";
    else
        cout << "Pessoa menor de idade\n";

    return 0;
}
```

Exemplo 9

- Faça uma função chamado `maisVelha` que receba duas `struct Pessoa` e retorne a `struct Pessoa` que contém a pessoa mais velha.

Exemplo 9

```
#include <iostream>

using namespace std;

struct Pessoa {
    string nome;
    int idade;
};

struct Pessoa maisVelha(struct Pessoa p1, struct Pessoa p2)
{
    if (p1.idade > p2.idade)
        return p1;
    return p2;
}

int main()
{
    struct Pessoa alguem = {"Fulano", 48};
    struct Pessoa outro = {"Sicrano", 35};

    cout << maisVelha(alguem, outro).nome << " eh o mais velho.\n";

    return 0;
}
```

Funções e ponteiros para estruturas

- É melhor passar ponteiros
 - ▶ Para evitar uma sobrecarga na pilha, é preferível passar ponteiros para estruturas como parâmetros de funções.
 - ▶ Isso evita que a estrutura seja toda copiada na pilha cada vez que a função é chamada.

Exemplo 10

```
#include <iostream>

using namespace std;

struct Pessoa {
    string nome;
    int idade;
};

int maiorIdade(struct Pessoa *p)
{
    if ((*p).idade < 21)
        return 0;
    return 1;
}

int main()
{
    struct Pessoa alguem = {"Fulano", 28};

    if(maiorIdade(&alguem))
        cout << alguem.nome << " eh maior!\n";
    else
        cout << alguem.nome << " eh menor!\n";

    return 0;
}
```

Exemplo 11

```
#include <iostream>

using namespace std;

struct Pessoa {
    string nome;
    int idade;
};

struct Pessoa *maisVelha(struct Pessoa *pessoas, int tam)
{
    int i;
    struct Pessoa *pAux=pessoas;

    for(i=0; i<tam; i++)
        if ((pessoas+i)->idade > pAux->idade)
            pAux = pessoas + i;
    return pAux;
}

int main()
{
    struct Pessoa vet[3] = {"Fulano", 48}, {"Sicrano", 35}, {"Beltrano", 78}};
    cout << maisVelha(vet,3)->nome << " eh o mais velho.\n";

    return 0;
}
```


Perguntas?