

Laboratório

Algoritmos e


Estruturas de dados I

Funções

Ramon – 7º Período – Engenharia da Computação

Roteiro

- **Definição**
- **Protótipo de uma Função**
- **Parâmetros de uma Função**
- **Recursividade**
- **Exemplos**
- **Exercícios**


Instituto Nacional de Telecomunicações

Funções

www.inatel.br

Definição

Uma função, sub-rotina ou sub-algoritmo é um conjunto de comandos agrupados em um bloco que possui um nome e pode ser ativado, ou seja, é um miniprograma dentro de um programa.


Instituto Nacional de Telecomunicações

Funções

www.inatel.br

Estrutura de uma Função


```
tipo nome_da_função(parâmetros)
{
    //comandos
}
```

tipo – É o tipo do dado de retorno da função.
ex: int, float, double,...;

nome_da_função – É o nome pela qual será possível chamar a função.

parâmetros – valores que serão passados para a função.

www.inatel.br



Instituto Nacional de Telecomunicações

Funções

Exemplos

```
void soma(int a, int b)
{
    int c;

    c = a + b;
    cout << c;
}
```

Exibe o resultado da soma de 2 números inteiros passados para a função.

```
float soma(float a, float b)
{
    return (a+b);
}
```

Retorna o resultado da soma de 2 números reais.

www.inatel.br



Instituto Nacional de Telecomunicações

Funções

Protótipo de uma Função

A linguagem C/C++ requer que a função seja declarada antes de seu uso, esta declaração é chamada de **protótipo**.

tipo nome_da_função(parâmetros);

Exemplos

```
float soma(float a, float b);

int main()
{
    float c,a=10,b=20;

    c = soma(a,b);
    cout << c;
}

float soma(float a, float b)
{
    return (a+b);
}
```

Parâmetros de uma Função

São valores passados para uma função, funcionam dentro da função como qualquer outra variável.

Uma função pode possuir: **passagem de parâmetros por valor**, **passagem de parâmetros por referência**, ou ainda, **não possuir parâmetro**.

Parâmetros de uma Função

•Passagem de Parâmetro por Valor

Também chamada de passagem por cópia, a função recebe apenas uma cópia dos valores passados a ela. Se o valor da variável for alterado, a mudança afeta apenas a função.

```
float soma(float a, float b);

int main()
{
    float c,a=10,b=20;

    c = soma(a,b);
    cout << c;
}
```

```
float soma(float a, float b)
{
    return (a+b);
}
```

Parâmetros de uma Função

•Passagem de Parâmetro por Referência

Também chamada de passagem por endereço, a função recebe o endereço de memória do valor do parâmetro. Se o valor da variável for alterado, a mudança afeta todas as funções que possuírem a variável.

```
float soma(float &a, float &b);

int main()
{
    float c,a=10,b=20;

    cout << a << " + " << b;
    c = soma(a,b);
    cout << " = " << c << endl;
    cout << "\n" << a; //a = 50
    cout << "\n" << b; //b = 100
}
```

```
float soma(float &a, float &b)
{
    float result = a + b;
    a = 50;
    b = 100;
    return result;
}
```

Parâmetros de uma Função

•Sem Parâmetro

Não há valores passados à função.

```
float soma();

int main()
{
    cout << "10 + 10 = " << soma();
}

float soma()
{
    float a=10,b=10;
    return (a+b);
}
```

Função Recursiva

Uma função recursiva é uma função que chama a si mesma.

```
#include<iostream>
#include<cstdlib>
using namespace std;

void fatorial(int fat,int cont) {
    if(cont != 1) {
        cont--;
        fat *= cont;
        fatorial(fat,cont);
    }
    else
        cout << "\nFatorial eh " << fat;
}

int main() {
    int fat = 1;

    cout << "Entre com um valor inteiro: ";
    cin >> fat;
    fatorial(fat,fat);
    cin.ignore();
    cin.get();
}
```

Exemplos

1. Faça um programa que tenha uma função que receba 3 valores como parâmetro e retorne a média aritmética.
2. Elabore um programa que tenha uma função sem retorno que verifica se o parâmetro é um número positivo. Se não for a função deve mudar o valor da variável para o módulo deste valor.
3. Desenvolva um programa que possua uma função que exiba um nome passado por parâmetro.