Algoritmos e Estruturas de Dados

Disciplina 301477

http://www.nickgentry.com/

Programa de Pós-graduação em
Computação Aplicada

Prof. Alexandre Zaghetto
http://alexandre.zaghetto.com
zaghetto@unb.br

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Módulo 09
Subalgoritmos
(Funções)

1. Funções

• Frequentemente temos de desenvolver programas para resolver problemas que necessitam de algoritmos extensos e complexos.

• Isso costuma implicar em códigos mais difíceis de ler e em repetição de trechos de código ao longo do programa.

Em programação podemos dividir os **algoritmos** em **subalgoritmos** menores e de mais fácil compreensão.

• Com isso, dividimos um problema difícil em vários problemas mais fáceis, juntando tudo no final e formando uma solução completa.

1. Funções

• Até agora, em todos os programas que criamos, codificamos uma única função: a função ***main***().

• Entretanto, em todos eles, diversas funções foram utilizadas: ***system***()***,*** ***printf***(), ***scanf***(), ***getch***(), ***putch***(), ***sqrt***(), ***pow***() etc.

• Essas funções estão disponíveis no sistema através de bibliotecas que acompanham o compilador C.

• Mas podemos definir nossas próprias funções e utilizá-las da mesma maneira.

2. Definindo Funções

```
tipo <nome> (<parâmetros>) {
    <Declarações de variáveis>
    <comandos>
}
```

• ***tipo*** refere-se ao tipo de resposta que a função devolve e deve ser ***void*** (vazio) se a função não tem valor de resposta;

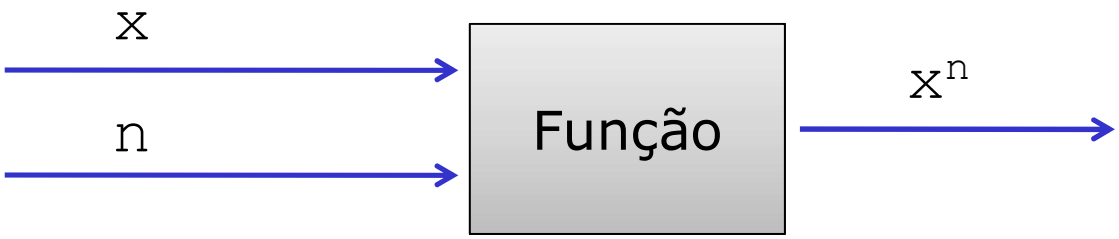
• ***nome*** é o identificador da função no resto do programa;

• ***parâmetros*** é uma lista de variáveis que representam valores de entrada para a função e deve ser ***void*** caso não haja valores de entrada;

• Dentro do corpo da função, a primeira seção é destinada à declaração das variáveis e a segunda, aos comandos.

2. Definindo Funções

- Função que tem valor de resposta e que recebe argumentos **por valor** ao ser chamada.



2. Definindo Funções

Retorna um ***float.***

Recebe um ***float*** e um ***int*.**

```
float potencia(float x, int n){
    float pot;

    return pot;
}
```

2. Definindo Funções

```
float potencia(float x, int n){

    float pot = 1;
    int i;
    for(i=0; i<n; i++) pot = pot*x;
    return pot;
}
```

2. Definindo Funções

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

float potencia(float, int);
```

```
float potencia(float x, int n){

    float pot = 1;
    int i;
    for(i=0; i<n; i++) pot = pot*x;
    return pot;
}
```

2. Definindo Funções

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

float potencia(float, int);

int main(){

    float resultado;

    resultado = potencia(5,2);
    printf("%f \n", resultado);


    return 0;
}
float potencia(float x, int n){

    float pot = 1;
    int i;
    for(i=0; i<n; i++) pot = pot*x;
    return pot;
}
```

"No fim das contas, porém, o fato é que *nós educamos a nós mesmos.* Nós aprendemos, antes de tudo, decidindo aprender, assumindo um compromisso com a aprendizagem, que, por sua vez, gera concentração."

Salman Khan