

# **MINICURSO DE ALGORITMOS 2**

## **PET-SIMC**

**FUNÇÕES**

PET de Sistemas de Informação do Campus Monte Carmelo

@petsimc



PET de Sistemas de Informação do Campus Monte Carmelo

@petsimc

# CRONOGRAMA:

1. Revisão com desafio
2. Funções
  - a. Conceito
  - b. Sintaxe
3. Exercícios



## Desafio

Desenvolver um programa em C que permita ao usuário verificar quantos números acertou em uma cartela de loteria, com a possibilidade de verificar múltiplas cartelas usando a mesma combinação sorteada.

### 1. Entrada

- Os 6 números oficiais do sorteio (entre 1 e 60).
- Os 6 números de sua cartela. (entre 1 e 60)

### 2. Saída

- O programa informa quantos números foram acertados.
- Após cada verificação, o programa pergunta se o usuário deseja:
  - Verificar outra cartela (usando o mesmo resultado do sorteio)
  - Sair do programa



## O que são funções

$$f(x)$$

Vocês já usaram funções... lembram quais??

@petsimc

## O que são funções

$$f(x)$$

Vocês já usaram funções... lembram quais??

**scanf, printf** são bons exemplos de função.

Uma característica interessante da função é **usá-la sem necessariamente saber o que está acontecendo internamente**. Como os exemplos acima.

## O que são funções

$$f(x)$$

Na matemática e na programação, as funções são parecidas, você passa algo (**Parâmetro**) e recebe um resultado/resposta.

## Sintaxe:

Para declarar uma função, basta dar um nome a ela, indicar seus **parâmetros** e **retorno**:

```
1  int max(int a, int b){  
2      if(a>b) return a; // se (a>b) a função deve retornar o valor de "a"  
3  
4      // se a função chegou aqui é porque "a" não é maior que "b",  
5      return b; // logo b>=a e devemos retornar o valor de b  
6  }
```



## Sintaxe:

E como uso/chamo essa função no meu programa?

```
1  int main(){
2
3      int n, m; // declaro as variáveis "n" e "m", os dois números a serem comparados
4
5      scanf("%d %d", &n, &m); // leio os valores de "n" e "m" na tela
6
7      printf("%d", max(n, m)); // imprimo o valor retornado pela função max
8
9      return 0;
10 }
```



## Sintaxe:

E se minha função não retornar nada? Então ela deve ser do tipo...

```
1  void nada(){  
2  |    printf("Nao retorno nada :>");  
3  }
```



## Ponto importante!!!

### 1. **Se a função for implementada ANTES da main:**

- Não é necessário declarar um protótipo antes
- O compilador já "conhece" a função quando encontra a chamada na main

### 2. **Se a função for implementada DEPOIS da main:**

- Você DEVE declarar um protótipo antes da main
- Caso contrário, o compilador não reconhecerá a função quando encontrar sua chamada

## Ponto importante!!!

### 1. Se a função for implementada ANTES da main:

```
#include <stdio.h>

// Função já implementada antes da main
int dobrar(int x) {
    return x * 2;
}

int main() {
    printf("Dobro de 5: %d\n", dobrar(5)); // Compilador já conhece a função
    return 0;
}
```

## Ponto importante!!!

### 2. Se a função for implementada DEPOIS da main:

```
#include <stdio.h>

// Protótipo da função (declaração antecipada)
int dobrar(int x);

int main() {
    printf("Dobro de 5: %d\n", dobrar(5)); // Compilador conhece pelo protótipo
    return 0;
}

// Implementação da função depois da main
int dobrar(int x) {
    return x * 2;
}
```



**Vamos praticar?!**

@petsimc



## Exercício 01 - Média escolar

Escreva uma função que receba 3 notas de um aluno (P1, P2, P3), e retorna a média aritmética das notas do aluno.

Depois modifique a função para calcular a média na função e mostrar o resultado **aprovado** se a média for maior ou igual a 6 e **reprovado** se for menor que 6, sem retornar nada.



## Exercício 02 - Bases e Expoentes

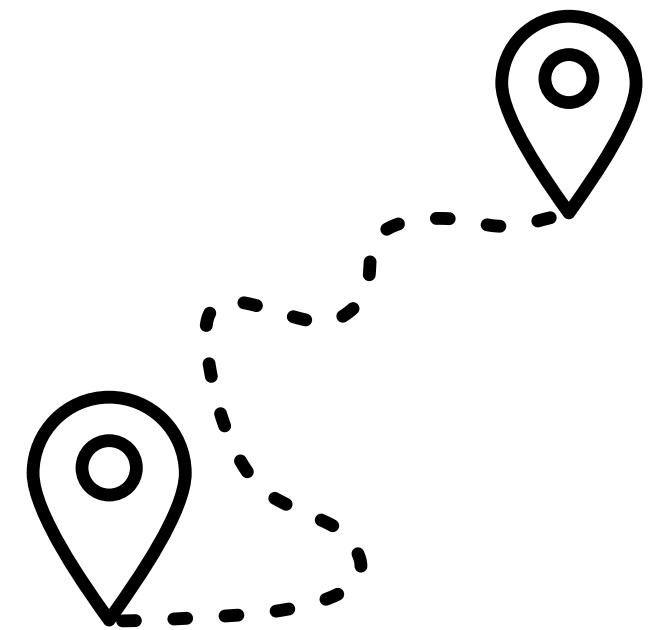
Escreva uma função `potencia(base, expoente)` que, quando chamada, retorna base elevada ao expoente. Por exemplo, `potencia(3, 4)` deve retornar 81. Assuma que expoente é um inteiro maior ou igual a 1.

$a^b$



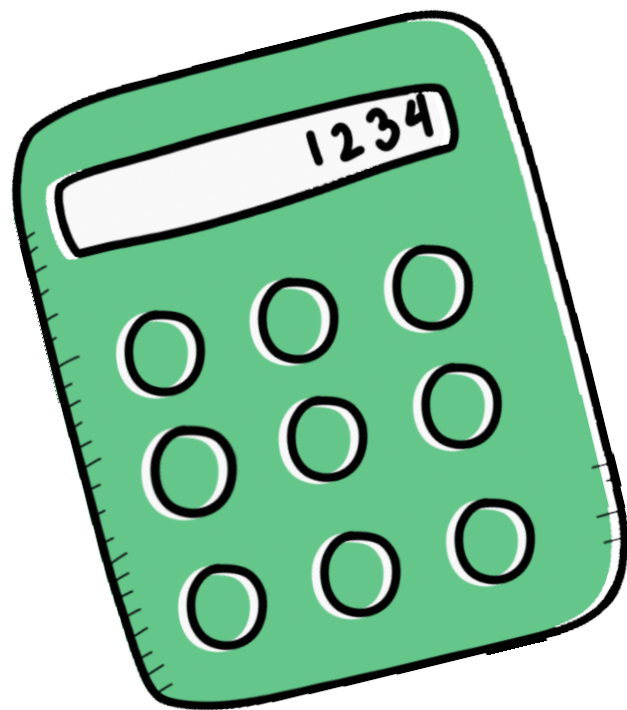
### Exercício 03 - Distância Euclidiana

Escreva uma função que calcule e retorne a distância entre dois pontos (  $x_1, y_1$ ) e (  $x_2, y_2$ ). Todos os números e valores de retorno devem ser do tipo float.



### Exercício 04 - Números

Faça uma função que recebe um valor inteiro e verifica se o valor é positivo, negativo ou zero. A função deve retornar 1 para valores positivos, -1 para negativos e 0 para o valor 0.



## CONTATO

Como entrar em contato em caso de dúvidas?



### E-mail

laviniabd@ufu.br  
nicolly.luz@ufu.br



### Sala do PET- 1BMC206

UFU - Monte  
Carmelo - Campus  
Araras



### Site do PET-SIMC

[petsimc.facom.ufu.br](http://petsimc.facom.ufu.br)



### Grupo do Wpp



@petsimc