

# SUB-ROTINAS

Introdução à Computação I

# Introdução

- Sub-rotinas, também chamadas de subprogramas, são blocos de instruções que realizam tarefas específicas
- O código de uma sub-rotina é carregado uma vez e pode ser executado quantas vezes forem necessárias
- Como o problema pode ser subdividido em pequenas tarefas, os programas tendem a ficar menores e mais organizados

# Introdução

- Os programas em geral são executados linearmente, uma linha após a outra, até o fim
- Entretanto, quando sub-rotinas são utilizadas é possível a realização de desvios na execução dos programas
- Estes desvios são efetuados quando uma função é chamada pelo programa principal

# Introdução

## ALGORITMO

DECLARE sal NUMÉRICO

LEIA sal

**aum**  $\leftarrow$  **calculo(sal)**

novo\_sal  $\leftarrow$  sal + aum

ESCREVA “Novo salário é: “, novo\_sal

FIM\_ALGORITMO

## **SUB\_ROTINA calculo (sal NUMÉRICO)**

DECLARE perc, valor NUMÉRICO

LEIA perc

valor  $\leftarrow$  sal \* perc / 100

RETORNE valor

**FIM\_SUB\_ROTINA calculo**

# Sub-rotinas em C

- Um importante recurso apresentado nas linguagens de programação é a modularização, onde um programa pode ser particionado em sub-rotinas bastante específicas
- A linguagem C possibilita a modularização por meio das funções
- Um programa escrito na linguagem C tem, no mínimo, uma função chamada **main**, por onde a execução começa

# Sub-rotinas em C

- Existem também muitas outras funções predefinidas na linguagem C, por exemplo: `gets()`, `sqrt()`, `getche()`, etc.
- Essas funções são adicionadas aos programas pela diretiva `#include`
- Além disso, o usuário também pode criar quantas funções quiser, dependendo do problema que estiver sendo resolvido pelo programa

# Sub-rotinas em C

- As funções às vezes precisam receber valores externos, chamados parâmetros, e também podem devolver algum valor produzido para o ambiente externo, denominado retorno
- Os parâmetros são representados por uma lista de variáveis colocadas dentro de parênteses, logo após o nome da função
- Caso haja retorno, a última linha da função deverá incluir o comando **return**, seguido do valor ou variável que será devolvido a quem chamou a função

# Sub-rotinas em C

- O tipo do valor retornado deverá ser exatamente igual ao tipo informado antes do nome da função
- Caso não haja retorno deverá ser digitada a palavra **void**
- Os tipos de funções são:
  - ▣ sem passagem de parâmetros e sem retorno
  - ▣ sem passagem de parâmetros e com retorno
  - ▣ com passagem de parâmetros e sem retorno
    - passagem de parâmetros por valor
    - passagem de parâmetros por referência
  - ▣ com passagem de parâmetros e com retorno
    - passagem de parâmetros por valor
    - passagem de parâmetros por referência



# Funções sem passagem de parâmetros e sem retorno

- Exemplos
- Exemplo clrscr();
  - ▣ **void clrscr(void);**

# Funções sem passagem de parâmetros e com retorno

- Exemplos
- Exemplo getch();
  - ▣ **int getch(void);**

# IMPORTANTE

- Dentro das sub-rotinas pode ocorrer a declaração de variáveis, chamadas **variáveis locais**
- Elas recebem esse nome uma vez que podem ser utilizadas apenas dentro da sub-rotina
- Quando a execução da mesma chega ao fim, essas variáveis são destruídas e seus conteúdos são perdidos

# IMPORTANTE

- Variáveis declaradas fora de qualquer sub-rotina são chamadas de **globais**
- Elas recebem esse nome uma vez que em qualquer ponto do programa, incluindo as sub-rotinas, podemos utilizá-las
- São destruídas quando a execução do programa chega ao fim
- **Não se aconselha a utilização excessiva de variáveis globais por tornar difícil a manutenção e a busca por erros nos programas. Além disso, as sub-rotinas acabam não ficando “genéricas”, dificultando sua reutilização (as mesmas são projetadas para uma tarefa específica)**

# Exercícios

- Faça uma sub-rotina que leia cinco valores inteiros, determine e mostre o maior e o menor deles. [sem parâmetro, sem retorno]
  - ▣ Os valores deverão ser locais a função
- Faça uma sub-rotina que calcule  $X$  elevado à  $Y$  (sem usar a função “pow”). Para tanto, leia  $X$  e  $Y$  e exiba o resultado da chamada da sub-rotina na tela (na função main). Chamar a função até que o usuário não deseje mais efetuar nenhum cálculo. [sem parâmetro, com retorno]
  - ▣ Os valores deverão ser locais a função