

Programação Básica de Computadores

Jordana S. Salamon
jssalamon@inf.ufes.br

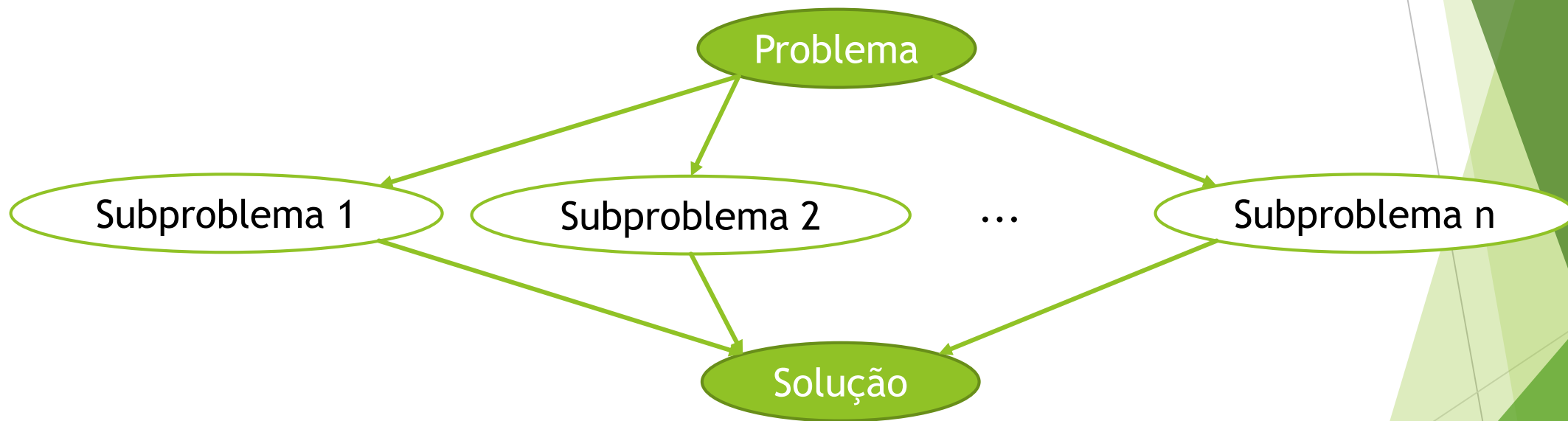
DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

$$I_i = \int_{x_i}^{x_{i+1}} f(x) dx$$

Funções em C

Motivação da utilização de Funções

- Dividir para conquistar



Motivação da utilização de Funções

- ▶ Subdivisão de algoritmos complexos
 - ▶ Facilidade de entendimento
- ▶ Estruturação de algoritmos
 - ▶ Facilidade de documentação e de detecção de erros
- ▶ Modularização de Sistemas
 - ▶ Facilidade de manutenção e reutilização
- ▶ Exemplos
 - ▶ Bibliotecas: math, string, stdio, etc
 - ▶ Parte do programa que é repetida várias vezes
 - ▶ Verificar se o número é primo
 - ▶ Calcular média
 - ▶ Fatorial



Sintaxe

- ▶ Tipo do retorno
 - ▶ Define o tipo a ser retornado após o processamento
- ▶ Nome da função
 - ▶ Semelhante ao nome do algoritmo em pseudo-código
- ▶ Tipo e nome do parâmetro

```
<tipo_retorno> <nomeFuncao> (<parametros>) {  
    <codigos>  
    return ...;  
}
```

Exemplo

```
#include<stdio.h>

int fatorial(int n){
    int f,i;
    f = 1;
    for(i = 1; i <=n; i++){
        f = f * i;
    }
    return f;
}

int main(){
    int fat;
    fat = fatorial(5);
    printf("Fatorial de 5 é: %d",fat);
    fat = fatorial(10);
    printf("Fatorial de 10 é: %d",fat);
    return 0;
}
```



Sintaxe

- ▶ Procedimentos

- ▶ Não retornam um valor
- ▶ void

- ▶ Funções

- ▶ Retornam um valor como resultado do processamento
- ▶ int, float, char, ...

Variáveis Globais e Locais

- ▶ Declaração de variáveis dentro da função

```
void função(int n){  
    int x;  
}
```

- ▶ Acesso a variáveis globais e locais

- ▶ Variável definida em uma função não é acessível por outra

- ▶ Passagem de parâmetro por cópia

- ▶ Alteração do parâmetro não interfere no valor original

Variáveis Locais

Só existem dentro da função fatorial

```
#include<stdio.h>

int fatorial(int n){
    int fat,i;
    fat = 1;
    for(i = 1; i <=n; i++){
        fat = fat * i;
    }
    return fat;
}

int main(){
    int fat;
    fat = fatorial(5);
    printf("Fatorial de 5 é: %d",fat);
    fat = fatorial(10);
    printf("Fatorial de 10 é: %d",fat);
    return 0;
}
```

Só existe dentro da função main

Função Main()

- ▶ Também é uma função, mas especial:
 - ▶ Chamada pelo sistema operacional (SO) quando executamos nosso programa
- ▶ Podemos utilizar a forma mais simples
 - ▶ `main(){ }`
- ▶ Forma completa
 - ▶ `int main(int argc, char **argv){ }`
- ▶ Retorno indica sucesso ou falha para o SO
 - ▶ Sucesso: `= 0`
 - ▶ Erro: `> 0`
- ▶ Parâmetros
 - ▶ Lista de parâmetros (`argv`)
 - ▶ Quantidade de parâmetros (`argc`)

Vetor em Funções

- Dentre o parâmetros que podemos passar para uma função, a linguagem C nos permite que vetores sejam passados de parâmetro. Exemplo:

```
#include <stdio.h>

void imprimirVetor(int v[10]){
    int i;
    printf("Vetor: ");
    for(i=0;i<=9;i++){
        printf("%d ", v[i]);
    }
}

int main(){
    int vet[10], i;
    for(i=0;i<=9;i++){
        printf("Informe o num da pos %d: ", i);
        scanf("%d", &vet[i]);
    }
    imprimirVetor(vet);
}
```

Vetor em Funções

- É possível também que o parâmetro seja um vetor de tamanho dinâmico, para isso é necessário que uma variável inteira guardando o tamanho do vetor também seja passada por parâmetro para a função. Exemplo:

```
#include <stdio.h>

void imprimirVetor(int n, int v[n]){
    int i;
    printf("Vetor: ");
    for(i=0;i<=n-1;i++){
        printf("%d ", v[i]);
    }
}

int main(){
    int vet[10], i;
    for(i=0;i<=9;i++){
        printf("Informe o num da pos %d: ", i);
        scanf("%d", &vet[i]);
    }
    imprimirVetor(10, vet);
}
```

Exercícios

1. Faça um procedimento (função com retorno vazio) que recebe a um número inteiro imprima se ele é par ou ímpar. No programa principal leia indeterminados números positivos (até que seja informado um número negativo) e informe para cada um desse número se ele é par ou ímpar.
2. Faça um programa que leia um número N (inteiro) e calcule e imprima o fatorial de N. Para isso desenvolva uma função para calcular o fatorial de um número.
3. Faça um programa em C que leia um número N (inteiro) e calcule e imprima a quantidade de dígitos que o número N possui. Para isso desenvolva uma função para calcular a quantidade de dígitos de um número.
4. Faça uma função que verifique se um numero é primo. A função deve retornar um valor lógico (0 - falso e 1 - verdadeiro). No programa principal leia indeterminados números inteiros positivos (até que o usuário digite -1) e escreva se cada um dos números é primo ou não.

Exercícios

5. Faça uma função que calcule a série de Fibonacci. No programa principal leia um número N e imprima o valor da sequência na posição N. Série de Fibonacci = 1, 1, 2, 3, 5, 8, 13, 21, 34...
6. Fazer um programa que leia 20 números reais e os armazene em um vetor. Após isso, chame um procedimento que imprima a quantidade de números que são maiores que a média desses 20 números.
7. Faça um procedimento que imprima os valores de um vetor. No programa principal leia um valor N e logo em seguida preencha um vetor com N números inteiros positivos lidos do usuário. Ao fim imprima os valores do vetor.
8. Faça uma função que receba como parâmetro um caractere que represente uma operação (+, -, / ou *) e um vetor de números. Aplique de forma crescente a operação aos valores do vetor e retorne o resultado. Ex.: operação = +, vetor = [1,2,3,4], retorna: 10

That's all Folks!



nemo