

Algoritmos de Programação 2

Professor: José Eurípedes Ferreira de Jesus Filho
jeferreirajf@gmail.com

Universidade Federal de Jataí – UFJ

O que é alocação estática? O que
é alocação dinâmica?

Quando faz sentido utilizar
alocação dinâmica?

O que iremos aprender nessa
aula?

Que tal resolvermos um exercício
antes?

Exercícios

- Desenvolva o código em C de um programa que recebe um valor inteiro n e imprima na tela o valor da seguinte fórmula:

$$1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

Vamos falar de um conceito
necessário para esta aula...

Introdução

- É comum que um mesmo trecho de código precise ser repetido várias vezes dentro de diferentes lugares em um programa.
- É muito mais fácil construir programas pequenos e simples e a partir deles chegar a um programa grande do que construir um programa grande e complexo diretamente.
- É muito mais fácil entender e manter um programa pequeno do que um programa grande.

Procedimentos e Funções

- Um procedimento é um subprograma.
 - ✓ Possui um nome identificador.
 - ✓ Possui um conjunto de instruções a serem executadas.
 - ✓ Possui variáveis próprias.
 - ✓ É possível enviar informações na invocação de um procedimento.
 - ✓ Ao finalizar a execução de todas as instruções, o programa volta do ponto onde o procedimento foi invocado.
- Exemplo:
`printf("Essa é uma invocação de um procedimento");`

Procedimentos e Funções

- Uma função é um subprograma que retorna um valor.
 - ✓ Possui um nome identificador.
 - ✓ Possui um conjunto de instruções a serem executadas.
 - ✓ Possui variáveis próprias.
 - ✓ É possível enviar informações na invocação de uma função.
 - ✓ Ao finalizar a execução de todas as instruções, o programa volta do ponto onde a função foi invocada retornando um valor.
- Exemplo:

```
double x = sqrt(9);
```

Procedimentos e Funções

- Vantagens:
 - ✓ Dividir os programas em blocos (subprogramas).
 - ✓ Evitar repetição de código.
 - ✓ Evitar programas muito grandes.
 - ✓ Facilitar manutenção dos programas.
 - ✓ Facilitar leitura.
 - ✓ Reusabilidade.
 - ✓ Modularidade.
 - ✓ Facilitar testes.

Procedimentos e Funções

- Procedimento em C:

```
void identificador(tipo parametro1, tipo parametro2, ... ){  
    comandos;  
}
```

- O conjunto de parâmetros pode ser vazio.

Procedimentos e Funções

- Função em C:

```
tipo identificador(tipo parametro1, tipo parametro2, ... ){  
    comandos;  
    return valor;  
}
```

- O conjunto de parâmetros pode ser vazio.

Procedimentos e Funções

- Desenvolva o código em C utilizando funções e procedimentos de um programa que recebe um valor inteiro n e imprima na tela o valor da seguinte fórmula:

$$1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

Procedimentos e Funções

- Decompor um programa em subprogramas
 - ✓ Facilita a leitura, o entendimento, os testes, a reusabilidade e a manutenção.
 - ✓ Possibilita a construção de bibliotecas.
 - ✓ Encontrar a melhor decomposição requer experiência.

Procedimentos e Funções

- Diferentes funções podem possuir variáveis com o mesmo identificador.
- Uma variável declarada dentro de uma função “morre” no final da execução da função.
- É importante entender o escopo de cada variável.

Passagem de parâmetros

- Existem dois tipos de passagem de parâmetros para uma função
 - ✓ **Valor**
 - ✓ Fornece uma cópia dos parâmetros para a função chamada
 - ✓ Qualquer modificação nos parâmetros dentro da função não altera os valores das variáveis fora da função
 - ✓ **Referência**
 - ✓ É passado o ENDEREÇO dos parâmetros para a função chamada
 - ✓ Qualquer modificação nos parâmetros em questão alteram os valores das variáveis fora da função
- **Evitar** passagens por **referência** é uma boa prática de programação.

Sumarizando

- **PROCEDIMENTOS**

- ✓ Subprograma sem retorno de valor.
- ✓ Usado para reduzir tamanhos de códigos, evitar repetições e facilitar reusabilidade e manutenção.

- **FUNÇÕES**

- ✓ Subprograma com retorno de valor.
- ✓ Mesma utilização que os procedimentos.

- **PARÂMETROS**

- ✓ Podem ser passados por valor ou por referência.
- ✓ Passagem por valor faz uma cópia dos parâmetros.
- ✓ Passagem por referência passa o endereço dos parâmetros e permite alteração dos valores.

Exercícios

- Refaça os programas que você fez até agora utilizando funções e procedimentos.