

Algoritmos de Programação I

Professor: José Eurípedes Ferreira de Jesus Filho

jeferreirajf@gmail.com

Universidade Federal de Jataí – UFJ

O que vimos até agora?

O que iremos aprender?

Estruturas de repetição

- Uma **instrução de repetição** permite ao programador especificar que um programa deve **repetir uma ou mais ações** enquanto alguma condição permanecer verdadeira.
- Também podemos chamar uma repetição de **laço** ou **loop**.
- Se a **condição** da repetição **nunca se tornar falsa**, de alguma forma, as ações da repetição serão executadas infinitamente (**loop infinito**).

Estruturas de repetição - Enquanto

- Em C, a instrução **ENQUANTO** obedece a seguinte sintaxe:

```
WHILE(condição){  
    instruções;  
}
```

- As instruções dentro do **ENQUANTO** serão executadas até que a condição avaliada seja falsa.
- O teste da condição é no início.

Exercício

- Desenvolva o código em C de um programa que soma números inteiros informados pelo usuário até que um número negativo seja informado e então imprime o resultado da soma na tela.

Estruturas de repetição – Faça/Enquanto

- A instrução **FAÇA/ENQUANTO** também é uma instrução de repetição similar ao **ENQUANTO**.
- Contudo, o bloco de instruções dentro do laço é executado e somente depois a condição é avaliada.
- O teste da condição é no final.

Estruturas de repetição – Faça/Enquanto

- Em C, a instrução **FAÇA/ENQUANTO** obedece a seguinte sintaxe:

```
DO {  
    instruções;  
} WHILE (condição);
```

- As instruções dentro do **laço** será executada ao menos uma vez. Além disso, as instruções serão executadas até que a condição seja falsa.

Exercício

- Desenvolva o código em C de um programa que imprime na tela um menu onde o usuário poderá escolher entre: (1) Opção 1, (2) Opção 2, (3) Opção 3, (4) sair. Caso a opção (4) seja escolhida o programa deve encerrar. Caso qualquer outra opção seja escolhida o programa deve imprimir na tela o menu novamente.

Estruturas de repetição – Para

- A instrução **PARA** também é uma instrução de repetição similar ao **ENQUANTO**.
- Contudo, a condição de parada geralmente está associada a uma contagem.
- A condição é testada no início.

Estruturas de repetição – Para

- Em C, a instrução **PARA** obedece a seguinte sintaxe:

```
FOR(inicialização contador; condição; incremento){  
    instruções;  
}
```

- As instruções dentro do **PARA** serão executadas até que a condição avaliada seja falsa.
- O teste da condição é no início.

Exercício

- Desenvolva o código em C de um programa que solicita um número inteiro n do usuário e então imprime na tela o resultado da somatória de todos os números de $0, 1, 2, \dots, n$, como a fórmula a seguir:

$$\sum_{i=0}^n i$$

Exercício

- Desenvolva o código em C de um programa que solicita um número inteiro n do usuário e então imprime na tela a somatória de todos os números pares inteiros no intervalo fechado $[1, n]$.

Exercício

- Desenvolva o código em C de um programa em que o usuário informa a nota de n alunos e o programa conta quantos alunos foram **aprovados** (nota ≥ 6) e quantos alunos foram **reprovados** (nota < 6).

Exercício

- Desenvolva o código em C de um programa que recebe uma variável inteira n e imprime na tela o valor de $n!$

Exercício

- Desenvolva um programa que imprime na tela “branco” e “cinza” alternadamente n vezes.

Exercício

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

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

Exercício

- Desenvolva um programa em C que calcula o resultado da seguinte fórmula:

$$\sum_{i=1}^n i + \prod_{i=1}^{\frac{n}{2}} \sqrt{i}$$

Exercício

- Desenvolva um programa em C que lê um número do teclado imprime na tela todos os divisores do número.

Exercício

- Desenvolva um programa em C que lê um número do teclado e verifica se ele é primo ou não.

Exercício

- Desenvolva um programa em C que calcule a seguinte fórmula:

$$f(x) = \begin{cases} 1, & x = 1 \\ 2 + f(x - 1), & x > 1 \end{cases}$$

Exercício

- Faça um programa em C que simule uma calculadora simplificada. O programa deve exibir um menu que possibilite o usuário a escolher entre as operações de soma, subtração, multiplicação, divisão e sair. Caso a opção sair seja escolhida, o programa encerra. Caso qualquer outra opção seja escolhida, o programa irá pedir ao usuário dois operandos e realizar a respectiva operação.