



Universidade Positivo

Algoritmos de Programação

Aula 13

Prof.^a Mariane Cassenote

mariane.cassenote@up.edu.br

Conteúdo



- Retomada das estruturas de repetição **enquanto/while** e **faca...enquanto/do...while**
- Estrutura de repetição **para/for**
- Estruturas de repetição **aninhadas**

Ao final do componente curricular será possível utilizar estruturas de repetição simples e aninhadas para resolver problemas algorítmicos.

Estruturas de Repetição – Síntese

Em pseudocódigo:

```
enquanto (<condição>) {  
    |  
    <bloco de comandos>  
    |  
}
```

Em linguagem C:

```
while (<condição>) {  
    |  
    <bloco de comandos>  
    |  
}
```

Em pseudocódigo:

```
faca {  
    |  
    <bloco de comandos>  
    |  
} enquanto (<condição>)
```

Em linguagem C:

```
do {  
    |  
    <bloco de comandos>  
    |  
} while (<condição>);
```

Estruturas de Repetição



Ao analisarmos o que ocorre nos laços de repetição estudados até agora, percebemos que:

- Ocorre a **inicialização de uma variável** envolvida na condição que controla o número de repetições
- A **condição que envolve essa variável é testada** (antes ou depois da execução do bloco de comandos da repetição, dependendo da estrutura utilizada)
- Dentro do laço de repetição ocorre uma **atualização no valor dessa variável** de forma a prever o encerramento do laço em algum momento (evitando loop infinito)

Para praticar



Escreva um programa que leia uma sequência de números inteiros até que o valor zero seja lido. Com esses valores, calcule a soma dos números ímpares e a média dos números pares.



Estruturas de Repetição

para / for

Estruturas de Repetição – para / for



- Indicado quando se conhece previamente a quantidade de repetições do laço a serem executadas
- O controle das repetições é feito por meio de um cabeçalho onde são definidos:
 - o nome da variável de controle (**contador**)
 - **valor inicial e valor final** da variável de controle
 - o valor do **incremento (ou decremento)** que a variável deve receber após cada repetição

Estruturas de Repetição – para / for

Sintaxe:

```
for (<inicialização>; <condição>; <incremento>) {  
    <bloco de comandos>  
}
```

Exemplo:

```
for (i = 1; i <= 5; i++) {  
    <bloco de comandos>  
} // saída: 1 2 3 4 5
```

Exemplo de for com
incremento de 1 (i++)

Estruturas de Repetição – para / for

Sintaxe:

```
for (<inicialização>; <condição>; <decremento>) {  
    <bloco de comandos>  
}
```

Exemplo:

```
for (i = 5; i >= 1; i--) {  
    <bloco de comandos>  
} // saída: 5 4 3 2 1
```

Exemplo de for com
decremento de 1 (i--)

Estruturas de Repetição – para / for



Elabore um programa que efetue o cálculo do fatorial do valor inteiro 5 e apresente o resultado dessa operação.

DICA: o fatorial é o produto (resultado da multiplicação) dos números inteiros positivos desde 1 até o limite informado, neste caso 5.

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

Estruturas de Repetição – para / for

```
// EM LINGUAGEM C – Calcule o fatorial de 5 (5!)
#include <stdio.h>

int main() {

    int i, fatorial;

    fatorial = 1;

    for (i = 1; i <= 5; i++) {
        fatorial = fatorial * i;
        printf("i = %d. fatorial = %d \n", i, fatorial);
    }

    printf("5! = %d", fatorial);

    return 0;
}
```

```
➤ ./main
i = 1. fatorial = 1
i = 2. fatorial = 2
i = 3. fatorial = 6
i = 4. fatorial = 24
i = 5. fatorial = 120
5! = 120➤
```

Estruturas de Repetição – para / for

// EM LINGUAGEM C – Calcule o fatorial de 5 (5!)

```
#include <stdio.h>
```

```
int main() {
```

```
    int i, fatorial;
```

```
    fatorial = 1;
```

```
    for (i = 1; i <= 5; i++) {  
        fatorial = fatorial * i;  
        printf("i = %d. fatorial  
    }
```

```
    printf("5! = %d", fatorial);
```

```
    return 0;
```

```
}
```

O que aconteceria se
inicializássemos “fatorial” como
0 (zero)?

```
➤ ./main  
i = 1. fatorial = 1  
i = 2. fatorial = 2  
i = 3. fatorial = 6  
i = 4. fatorial = 24  
i = 5. fatorial = 120  
5! = 120➤
```

Estruturas de Repetição – para / for

```
// EM LINGUAGEM C – Calcule o fatorial de 5 (5!)
#include <stdio.h>

int main() {

    int i, fatorial;

    fatorial = 0;

    for (i = 1; i <= 5; i++) {
        fatorial = fatorial * i;
        printf("i = %d. fatorial = %d \n", i, fatorial);
    }

    printf("5! = %d", fatorial);

    return 0;
}
```

```
> ./main
i = 1. fatorial = 0
i = 2. fatorial = 0
i = 3. fatorial = 0
i = 4. fatorial = 0
i = 5. fatorial = 0
5! = 0> □
```

Estruturas de Repetição – para / for

```
// EM LINGUAGEM C – Calcule o fatorial de 5 (5!)
#include <stdio.h>

int main() {

    int i, fatorial;

    fatorial = 1;

    for (i = 5; i >= 1; i--) {
        fatorial = fatorial * i;
        printf("i = %d. fatorial = %d \n", i, fatorial);
    }

    printf("5! = %d", fatorial);

    return 0;
}
```

```
➤ ./main
i = 5. fatorial = 5
i = 4. fatorial = 20
i = 3. fatorial = 60
i = 2. fatorial = 120
i = 1. fatorial = 120
5! = 120 ➤ □
```

Estruturas de Repetição – Em pseudocódigo

Sintaxe:

```
para (<inicialização>; <condição>; <incremento>) {  
    <bloco de comandos>  
}
```

Exemplo:

```
para (i = 1; i <= 5; i++) {  
    <bloco de comandos>  
} // saída: 1 2 3 4 5
```

Exemplo de for com
incremento de 1 (i++)

Estruturas de Repetição – Em pseudocódigo

Sintaxe:

```
para (<inicialização>; <condição>; <decremento>) {  
    <bloco de comandos>  
}
```

Exemplo:

```
para (i = 5; i >= 1; i--) {  
    <bloco de comandos>  
} // saída: 5 4 3 2 1
```

Exemplo de for com
decremento de 1 (i--)

Estruturas de Repetição – Em pseudocódigo

// EM PSEUDOCÓDIGO – Calcule o fatorial de 5 (5!)

```
programa {  
    funcao inicio() {  
  
        inteiro i, fatorial  
  
        fatorial = 1  
  
        para (i = 1; i <= 5; i++) {  
            fatorial = fatorial * i  
            escreva("i = ", i, ". fatorial = ", fatorial, "\n")  
        }  
  
        escreva("5! =", fatorial)  
    }  
}
```

```
i = 1. fatorial = 1  
i = 2. fatorial = 2  
i = 3. fatorial = 6  
i = 4. fatorial = 24  
i = 5. fatorial = 120  
5! =120
```

Programa finalizado. Tempo de execução: 37 ms

Estruturas de Repetição – Em pseudocódigo

// EM PSEUDOCÓDIGO – Calcule o fatorial de 5 (5!)

```
programa {  
    funcao inicio() {  
  
        inteiro i, fatorial  
  
        fatorial = 1  
  
        para (i = 1; i <= 5; i++)  
            fatorial = fatorial * i  
            escreva("i = ", i, "  
    }  
  
    escreva("5! =", fatorial)  
}
```

E se eu quiser fazer o laço
invertido (decremento)?

```
i = 1. fatorial = 1  
i = 2. fatorial = 2  
i = 3. fatorial = 6  
i = 4. fatorial = 24  
i = 5. fatorial = 120  
5! =120
```

Programa finalizado. Tempo de execução: 37 ms

Estruturas de Repetição – Em pseudocódigo

// EM PSEUDOCÓDIGO – Calcule o fatorial de 5 (5!)

```
programa {  
    funcao inicio() {  
  
        inteiro i, fatorial  
  
        fatorial = 1  
  
        para (i = 5; i >= 1; i--) {  
            fatorial = fatorial * i  
            escreva("i = ", i, ". fatorial = ", fatorial, "\n")  
        }  
  
        escreva("5! =", fatorial)  
    }  
}
```

```
i = 5. fatorial = 5  
i = 4. fatorial = 20  
i = 3. fatorial = 60  
i = 2. fatorial = 120  
i = 1. fatorial = 120  
5! =120
```

Programa finalizado. Tempo de execução: 37 ms

Estruturas de Repetição – Selecionando a melhor estrutura



- Quando se sabe exatamente quantas repetições serão realizadas, pode-se utilizar qualquer estrutura de repetição
- Quando não se sabe previamente quantas repetições serão executadas, indica-se o uso de **while** ou **do...while**
- Se existe a possibilidade de o bloco de comandos da repetição não precisar ser executado, indica-se o uso de **while**
- Se o bloco de comandos da repetição precisa necessariamente ser executado pelo menos uma vez, utilizar **do...while**

Para praticar



1. Desenvolver um algoritmo que efetue a soma de todos os números ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 500.
2. Escreva um algoritmo que leia um valor inicial A e uma razão R e imprima uma sequência em progressão aritmética contendo 10 valores.
 - Por exemplo, com $A = 1$ e $R = 3$, tem-se “1 4 7 10 13 16 19 22 25 28”
3. Escreva um algoritmo que leia um valor inicial A e uma razão R e imprima uma sequência em progressão geométrica contendo 10 valores.
 - Por exemplo, com $A = 1$ e $R = 3$, tem-se “1.00 3.00 9.00 27.00 81.00 243.00 729.00 2187.00 6561.00 19683.00”



Universidade Positivo

Algoritmos de Programação

Aula 13

Prof.^a Mariane Cassenote

mariane.cassenote@up.edu.br