



INF 1005 – Programação I

Aula 07 – Estruturas de Repetição

Edirlei Soares de Lima
<elima@inf.puc-rio.br>

Estruturas de Repetição

- Diversos problemas somente podem ser resolvidos numericamente por um computador se o resultado de pequenas computações forem acumulados.
 - **Exemplo:** calcular o fatorial de um número.
- Precisamos de mecanismos que nos permitam requisitar que um conjunto de instruções seja repetidamente executado, até que uma determinada condição seja alcançada.
- **Repetições são programadas através da construção de laços (ou ciclos).**

Estruturas de Repetição (WHILE)

- **Estruturas de repetição** são utilizadas para indicar que um determinado conjunto de instruções deve ser executado um número definido ou indefinido de vezes, ou enquanto uma condição não for satisfeita.
- Em C, uma das formas de se trabalhar com repetições é através do comando **while**:

```
...  
while (expressão_lógica)  
{  
    /* Bloco de comandos */  
}  
...
```

Enquanto a “**expressão_lógica**” for verdadeira, o “bloco de comandos” é executado!

Depois, a execução procede nos comandos subsequentes ao bloco while.

Estruturas de Repetição – Exemplo 1

- **Exemplo 1:**

“Crie um programa que escreva na tela todos os números entre 0 e 100”

```
#include <stdio.h>

int main (void)
{
    int x = 0;
    while (x <= 100)
    {
        printf("%d\n", x);
        x = x + 1;
    }
    return 0;
}
```

Estruturas de Repetição – Exemplo 2

- **Exemplo 2:**

“Fatorial de um número não-negativo”

$$n! = \prod_{i=1}^n i = n \times (n - 1) \times (n - 2) \times \cdots \times 3 \times 2 \times 1$$

```
int fatorial(int n)
{
    int f = 1;
    while(n > 1)
    {
        f = f * n;
        n = n - 1;
    }
    return f;
}
```

Estruturas de Repetição – Exemplo 3

- **Exemplo 3:**

“**MDC** (máximo divisor comum entre dois números inteiros positivos)”

Método de Euclides: efetua-se divisões sucessivas entre os dois números até obter resto zero. O máximo divisor comum entre os dois números iniciais é o último resto diferente de zero obtido.

MDC entre 42 e 24:

Passo	x	y	r
1	42	24	18
2	24	18	6
3	18	<u>6</u>	0

Estruturas de Repetição – Exemplo 3

```
int mdc(int x, int y)
{
    int r = x % y;
    while(r != 0)
    {
        x = y;
        y = r;
        r = x % y;
    }
    return y;
}
```

MDC entre 42 e 23:

Passo	x	y	r
1	42	23	19
2	23	19	4
3	19	4	3
4	4	3	1
5	3	<u>1</u>	0

Estruturas de Repetição – Exemplo 4

- **Exemplo 4:**

“Verificar se um Número é Primo”

Sabe-se que: Um número é dito primo se ele for divisível apenas pelo número 1 e pelo próprio número, sendo que 1 não é primo (2 é o primeiro número primo).

```
int primo(int n)
{
    int i;
    if (n < 2)
        return 0;
    i=2;
    while(i < n)
    {
        if (n%i == 0)
            return 0;
        i++;
    }
    return 1;
}

/* retorna 0 se n não for primo, 1 se for */
```


Estruturas de Repetição (FOR)

- Outra forma de se trabalhar com repetições é através do comando **for** – que é equivalente ao comando **while** com uma sintaxe mais compacta:

```
...  
for(expressão_inicial; expressão_lógica; expressão_atualização)  
{  
    /* Bloco de comandos */  
}  
...
```

Estruturas de Repetição – Exemplo 1

- **Exemplo 1:** “Escrever na tela os números entre 0 e 100”

```
#include <stdio.h>

int main (void)
{
    int x = 0;
    while (x <= 100)
    {
        printf("%d\n", x);
        x = x + 1;
    }
    return 0;
}
```



```
#include <stdio.h>

int main (void)
{
    int x;
    for (x = 0; x <= 100; x++)
    {
        printf("%d\n", x);
    }
    return 0;
}
```

Estruturas de Repetição – Exemplo 2

- **Exemplo 2:**

“Fatorial de um número não-negativo”

$$n! = \prod_{i=1}^n i = n \times (n - 1) \times (n - 2) \times \cdots \times 3 \times 2 \times 1$$

```
int fatorial(int n)
{
    int i;
    int f = 1;
    for(i=2; i<=n; i++)
    {
        f = f * i;
    }
    return f;
}
```

Estruturas de Repetição – Exemplo 3

- **Exemplo 3:**

“**MDC** (máximo divisor comum entre dois números inteiros positivos)”

```
int mdc(int x, int y)
{
    int r;
    for(r = x%y; r!=0; r = x%y)
    {
        x=y;
        y=r;
    }
    return y;
}
```

Estruturas de Repetição – Exemplo 4

- **Exemplo 4:**

“Verificar se um Número é Primo”

```
int primo(int n)
{
    int i;
    if (n < 2)
        return 0;
    for (i = 2; i < n; i++)
    {
        if (n % i == 0)
            return 0;
    }
    return 1;
}
```

Estruturas de Repetição (DO-WHILE)

- As estruturas **while** e **for** avaliam a expressão booleana que controla a execução do bloco de comandos no **início do laço**.
- A linguagem C oferece uma terceira construção de laços através do comando **do-while**:
 - A expressão booleana é avaliada no final do laço.
 - Isso significa que o bloco de comandos é executado pelo menos uma vez.

```
...  
do  
{  
    /* Bloco de comandos */  
}while(expressão_lógica);  
...
```

Tipos de Repetições

- **Repetição determinada**
 - Define-se o número de vezes que a repetição será executada antes da repetição começar.
- **Repetição indeterminada**
 - Cria-se uma condição de parada que é ativada durante o processo de repetição.

Repetição Indeterminada - Exemplo

- **Exemplo 5:**

“O professor de Prog I quer saber se os alunos estão gostando do curso. Crie um programa que permita aos alunos votarem em 1-Sim ou 2-Não. A votação deve terminar quando o voto 0 for digitado. Ao terminar a votação, o resultado deve ser mostrado.”


```
#include <stdio.h>

int main (void)
{
    int total_votos_sim = 0;
    int total_votos_nao = 0;
    int voto = -1;
    while (voto != 0)
    {
        printf("Digite o seu voto:\n1-Sim\n2-Nao\n0-Terminar Votacao\n");
        scanf("%d", &voto);
        if (voto == 1)
            total_votos_sim = total_votos_sim + 1;
        else if (voto == 2)
            total_votos_nao = total_votos_nao + 1;
    }
    printf("Total de votos Sim: %d\n", total_votos_sim);
    printf("Total de votos Nao: %d\n", total_votos_nao);
    return 0;
}
```

Exercícios

Lista de Exercícios 04 - Estruturas de Repetição

<http://www.inf.puc-rio.br/~elima/prog1/>