



BC-0005

Bases Computacionais da Ciência

**Aula 07 – Lógica de Programação:
Estruturas de repetição**

Laços (estruturas de repetição)

- Usado em situações em que é necessário repetir um determinado trecho de um programa, um determinado número de vezes.
- **Duas formas:**
 - Escrever o trecho quantas vezes for necessário, ou
 - Utilizar o conceito de Laços.

Laços (for)

```
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');
```



Estruturas de repetição: **For**

Laços (for)

```
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');  
disp('oi');
```

```
--> for i=1:10 disp('oi'); end;
```

Laços (for)

```
for variavel = inicio:incremento:fim  
    Instrucao_1;  
    Instrucao_2;  
    Instrucao_3;  
end
```

```
1 for i=1:2:10  
2     disp('oi');  
3 end  
4
```

```
1 for i=1:1:10  
2     r = rand();  
3     disp(r);  
4 end
```

(1) Imprimir intervalo

Crie uma função que permita mostrar a sequência de números, no intervalo $[x,y]$, considere $x < y$:

```
1 function imprimir_intervalo(x,y)
2     for i=x:y
3         disp(i);
4     end
5 endfunction
```

Teste com:

-->imprimir_intervalo(12,20)

-->imprimir_intervalo(-3,5)

(2) Soma de números

Crie uma função que permita somar a sequência de números inteiros, no intervalo $[x,y]$, considere $x < y$:

```
1 function soma = soma_intervalo(x,y)
2     soma = 0;
3     for i=x:y
4         soma = soma+i;
5     end
6 endfunction
```


(2) Soma de números

Crie uma função que permita somar a sequência de números inteiros, no intervalo $[x,y]$, considere $x < y$:

```
1 function soma = soma_intervalo(x,y)
2     soma = 0;
3     for i=x:y
4         soma = soma+i;
5     end
6 endfunction
```

```
-->soma_intervalo(0,4)
10.
```

```
-->soma_intervalo(-4,4)
0.
```

```
-->soma_intervalo(10,12)
33.
```

(3) Soma de números

Crie uma função (com nome somaP), em que dado um inteiro $n > 0$, permita somar a seguinte sequência:

$$1^2 + 2^2 + 3^2 + \dots + n^2$$

(3) Soma de números

```
1 function soma = somaP(n)
2     soma = 0;
3     for i=1:n
4         soma = soma+i^2;
5     end
6 endfunction
```

-->somaP(0)

0.

-->somaP(1)

1.

-->somaP(2)

5.

-->somaP(3)

14.

-->somaP(4)

30.

(4) Soma de números

Modifique a função somaP em que, além do número **n**, seja utilizado um outro número **k**, de tal forma que o calculo da seguinte somatória seja realizada:

$$1^k + 2^k + 3^k + \dots + n^k$$

(4) Soma de números

```
1 function soma = somaP(n,k)
2     soma = 0;
3     for i=1:n
4         soma = soma+i^k;
5     end
6 endfunction
```

-->somaP(10,4)
25333.

(5) Soma de números

Modifique a função **somaP** em que, além do número **n**, seja utilizado um outro número **k**, de tal forma que o calculo da seguinte somatória seja realizada:

$$k^1 + k^2 + k^3 + \dots + k^n$$

(5) Soma de números

```
1 function soma = somaP(n,k)
2     soma = 0;
3     for i=1:n
4         soma = soma+k^i;
5     end
6 endfunction
```

-->somaP(10,4)
1398100.



Estruturas de repetição: **While**

Soma de números

```
1 function soma = somaP(n,k)
2     soma = 0;
3     for i=1:n
4         soma = soma+k^i;
5     end
6 endfunction
```

```
1 function soma = somaP(n,k)
2     soma = 0;
3     i = 1;
4     while i<=n
5         soma = soma+k^i;
6         i = i+1;
7     end
8 endfunction
```

-->somaP(10,4)
1398100.

Estrutura: While

```
while condicao  
    Instrucao_1;  
    Instrucao_2;  
    ...  
    instrucao_n;  
end
```

O laço while (**enquanto**) repete uma sequência de instruções enquanto uma condição for satisfeita.

Útil quando não se sabe o número de iterações.

(1) Número de tentativas

```
1 function cara_menos_coroa()  
2     t = 0;  
3     while rand() > 0.5  
4         disp('cara');  
5     end  
6     disp('coroa - finaliza jogo');  
7 endfunction
```

-->cara_menos_coroa()

cara

cara

cara

coroa - finaliza jogo

(2) Fatorial

```
1 function mult = fatorial1(n)
2     mult = 1;
3     for i=1:n
4         mult = mult*i;
5     end
6 endfunction
```

(2) Fatorial

Crie uma função em que, dado um inteiro não-negativo **n**, seja possível determinar **n!**

(2) Fatorial

```
1 function mult = fatorial1(n)
2     mult = 1;
3     for i=1:n
4         mult = mult*i;
5     end
6 endfunction
```

(2) Fatorial

```
1 function mult = fatorial1(n)
2     mult = 1;
3     for i=1:n
4         mult = mult*i;
5     end
6 endfunction
```

```
1 function mult = fatorial2(n)
2     mult = 1;
3     i = 1;
4     while i<=n
5         mult = mult*i;
6         i = i+1;
7     end
8 endfunction
```

(3) Somatoria de 10mil termos

Faça uma função que calcula a soma:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{9999} - \frac{1}{10000}$$

Pelas seguintes maneiras:

- Adição de termos da esquerda para a direita;
- Adição de termos da direita para a esquerda;
- Adição separada dos termos positivos e dos termos negativos da esquerda para a direita;
- Adição separada dos termos positivos e dos termos negativos da direita para a esquerda.

Compare e discuta os resultados obtidos no computador.

(3) Somatoria de 10mil termos

```
1 function soma = somatoriaE()  
2     soma = 0;  
3     for i=1:10000  
4         soma = soma + (1/i) * (-1)^(i+1);  
5     end  
6 endfunction
```

Adição de termos da esquerda para a direita

(3) Somatoria de 10mil termos

```
1 function soma = somatoriaE2()  
2     soma = 0;  
3     i=10000;  
4     while i>=1  
5         soma = soma + (1/i) * (-1)^(i+1);  
6         i = i-1;  
7     end  
8 endfunction
```

Adição de termos da direita para a esquerda

Atividade 01

Crie uma função que permita somar apenas os números ímpares da sequência de inteiros contida no intervalo $[x,y]$, para $x \leq y$.

A função deve ser a seguinte:

```
function [soma] = soma_impares(x,y)  
...  
endfunction
```

Atividade 02

Dado um inteiro positivo **n**, crie uma função para calcular a seguinte soma:

A função deve ser a seguinte: **soma(n)**

$$\frac{1}{n} + \frac{2}{n-1} + \frac{3}{n-2} + \dots + \frac{n}{1}$$

Observe que você precisa apenas de um laço para fazer a soma

Atividade 03

Faça uma função **arctan** que recebe o número real $x \in [0,1]$ e devolve uma aproximação do arco tangente de x (em radianos) através da série:

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Nome da função: **arctan(x)**

Compare os resultados com a função `atan(x)` do `scilab`

Nota: Considere somente os 100 primeiros termos da série.

Nota2: Você não precisa