

# Aula 01 - Introdução à Linguagem C

## Programação: Estrutura-Repetição

IC2

Prof: Anilton Joaquim da Silva

[Anilton.ufu@outlook.com](mailto:Anilton.ufu@outlook.com)

# Linguagem C:

## Estrutura de Repetição

- Em certas situações é necessária a repetição de um conjunto de comandos. Em situações como esta, temos duas opções: ou copiamos e colamos todo o trecho que desejamos repetir, fazendo os ajustes necessários; ou utilizamos uma saída mais inteligente por meio de comandos especiais que permitem automatizar a repetição.

# Linguagem C:

## Comando **while** - enquanto

- Forma geral:

```
while (<condição>)  
{  
    ... //bloco de comandos a ser repetido  
}
```

- O bloco de comandos entre as chaves será **repetido** enquanto a condição dentro dos parênteses for verdadeira.

OBS: Se a condição lógica falhar inicialmente, i.e. se for falso, o bloco de comando não é executado nenhuma vez.

# Linguagem C:

## Exemplo - comando while

- #include <stdio.h>
- #include <stdlib.h>
  
- int main() // Programa que lê duas notas e imprime a média destas notas, de N alunos.
- {
- int numero, contador;
- float nota1, nota2, media;
  
- printf( "Programa de media de notas de N alunos.\n");
- printf( "Entre com o numero de alunos: ");
- scanf("%d", &numero);
- contador = 1;
- while (contador <= numero)
- {
- printf("\nEntre com 1 nota do aluno %d: ", contador);
- scanf("%f", &nota1);
- printf("Entre com 2 nota do aluno %d: ", contador);
- scanf("%f", &nota2);
- media = (nota1 + nota2)/2.0;
- printf("Media do aluno %d eh igual a: %f\n", contador, media);
- contador += 1; // contador = contador + 1; ou contador ++;
- }
- return 0;
- }

# Linguagem C:

## Exemplo - comando **while**

- `#include <stdio.h>`
- `#include <stdlib.h>`
  
- `int main() // Soma os números pares entre 1 e 999.`
- `{`
- `int n = 2, // primeiro número par maior do que 1`
- `soma = 0; // soma inicialmente zerada`
- `while (n < 999)`
- `{`
- `soma = soma + n;`
- `n = n + 2;`
- `}`
- `printf("\nO valor da soma eh %d", soma);`
  
- `printf("\n");`
- `return 0;`
- `}`

# Linguagem C:

## Comando **do - while**

- Forma geral:

```
do  
{  
    ... //bloco de comandos a ser repetido  
} while (<condição>);
```

- O bloco de comandos entre as chaves será **repetido** enquanto a condição dentro dos parênteses for verdadeira.

OBS: o bloco de comando é executado pelo menos uma vez.

# Linguagem C:

## Exemplo - comando do - while

- #include <stdio.h>
- int main() // **Programa que lê duas notas e imprime a média destas notas, de N alunos.**
- {
- int numero, contador;
- float nota1, nota2, media;
- printf( "Programa de media de notas de N alunos.\n");
- printf( "Entre com o numero de alunos: ");
- scanf("%d", &numero);
- contador = 1;
- do
- {
- printf("\nEntre com 1 nota do aluno %d: ", contador);
- scanf("%f", &nota1);
- printf("Entre com 2 nota do aluno %d: ", contador);
- scanf("%f", &nota2);
- media = (nota1 + nota2)/2.0;
- printf("Media do aluno %d eh igual a: %f\n", contador, media);
- contador += 1; // contador = contador + 1; ou contador ++;
- } while (contador <= numero);
- return 0;
- }

# Linguagem C:

## Exemplo - comando do - while

```
• #include <stdio.h>
• int main() // Programa que lê 10 números e imprime o maior
• {
•     int i = 0; // contador da qtde de numeros lidos
•     int maior, n;
•     do
•     {
•         printf("Entre com numeros: ");
•         scanf("%d", &n);
•         if (i == 0) // se for o primeiro numero lido
•         { //ele sera o maior
•             maior = n;
•         }
•         else // a partir do segundo
•         {
•             if(n > maior) // atualizo o maior
•             {
•                 maior = n;
•             }
•         }
•         i = i + 1; // i++; ou i +=1;
•     } while (i < 10);
•     printf("\nO maior numero lido eh: %d\n", maior);
•     return 0;
• }
```



# Linguagem C:

## Comando **for**

- Forma geral

```
for ( DI; C; I )  
{  
    ... //Bloco de comandos a serem repetidos  
}
```

- O comando **for** tem três partes em sua declaração, além dos comandos a serem repetidos.
  - **DI** – as variáveis podem ser **D**eclaradas e **I**niciadas. Variáveis já existentes também podem ter seus valores ajustados em **DI**;
  - **C** – define a **Condição** necessária à execução do bloco de comandos. Enquanto a condição for verdadeira, o bloco será executado.
  - **I** – comandos de modificação de variáveis, como Incremento e decremento, são colocados diretamente na declaração do **for**. O comando é executado ao final de cada iteração.

# Linguagem C:

## Comando **for**

- A execução do **for** segue os seguintes passos:
  1. **Iniciação** (execução de **DI**) – inicializa a variável controladora da repetição;
  2. **Avaliação** (teste da condição em **C**) – avalia o resultado lógico da comparação da variável controladora;
  3. **Execução** do bloco de comandos – é executado se a condição **for** verdadeira;
  4. **Incremento** – somente após a execução é que o função executa o incremento da variável controladora;
  5. **Volta** ao passo 2

# Linguagem C:

## Exemplo - comando **for**

- `#include <stdio.h>`
- `#include <stdlib.h>`
  
- `int main() // Soma os números pares entre 1 e 999.`
- `{`
- `int n;`
- `int soma = 0; // soma inicialmente zerada`
- `for (n = 2; n < 999; n = n + 2)`
- `{`
- `soma = soma + n;`
- `}`
- `printf("\nO valor da soma eh %d", soma);`
  
- `printf("\n");`
- `return 0;`
- `}`

# Linguagem C:

## Exemplo - comando for

- `#include <stdio.h>`
- `#include <stdlib.h>`
  
- `int main() // Programa que lê 10 números e imprime o maior.`
- `{`
- `int i; // contador da qtde de numeros lidos`
- `int maior, n;`
  
- `for (i = 0; i < 4; i = i + 1)`
- `{`
- `printf("Entre com um numero inteiro: ");`
- `scanf("%d", &n);`
- `if ((i == 0) || (n > maior)) // inicializo e atualizo o valor de maior`
- `{`
- `maior = n;`
- `}`
- `}`
- `printf("\nO maior numero lido eh: %d\n", maior);`
  
- `return 0;`
- `}`