

Estruturas de Repetição em C

Disciplina: PROGRAMAÇÃO II

Prof. Jean Eduardo Glazar
Curso de Sistemas de Informação
Campus Colatina-ES



INSTITUTO FEDERAL
Espírito Santo

WHILE

- ▶ A mesma lógica que no Python: repete a sequência de comandos **enquanto** a expressão for **verdadeira**.

```
while ( <expressão_condicional> )  
{  
    <sequência de comandos>  
}
```

Usa-se { e } para delimitar o bloco de execução da sequência de comandos.
A expressão condicional deve necessariamente estar entre parênteses.



WHILE - Exemplo

```
float nota;  
printf("Informe a nota: ");  
scanf("%f", &nota);  
while ( ( nota < 0 ) || ( nota > 10 ) ) {  
    printf("Nota inválida! Digite novamente: ");  
    scanf("%f", &nota);  
}
```

WHILE

- ▶ No comando WHILE, primeiro testa-se a condição e depois executa a sequência de comandos, por isso que no exemplo anterior foi preciso ler a nota antes do while e depois ler a nota depois do while. Dessa forma, a sequência de comandos pode não ser executada nenhuma vez.
- ▶ Essa repetição de comandos pode ser inconveniente.
- ▶ Para evitar isso, em C existe a estrutura de repetição:

do - while

DO-WHILE

- ▶ Nessa estrutura, a sequência de comandos é executada primeiro e somente depois que a condição é testada, por isso fica no final. Dessa forma, a sequência de comandos é executada pelo menos UMA vez.

```
do
{
    <sequência de comandos>
}
while ( <expressão_condicional> );
```

DO-WHILE - Exemplo

```
float nota;  
do {  
    printf("Digite a nota: ");  
    scanf("%f", &nota);  
} while ( ( nota < 0 ) || ( nota > 10 ) );
```

Note que o comando para leitura dos dados não foi duplicado.
Use essa estrutura quando você quiser repetir no mínimo uma vez.



FOR

- ▶ Essa estrutura de repetição é usada quando consigo contar quantas vezes a sequência de comandos será executada.
- ▶ Para isso, é necessário usar uma variável do tipo inteiro (**int**) que funcionará como contador. Com essa variável, define-se o valor **inicial**, o valor **final** e o **incremento**.

```
for ( <variável>=<valor_inicial> ; <condição de parada> ; <incremento> )  
{  
    <sequência de comandos>  
}
```

SEPARADOR

FOR – Exemplo 1

// Imprime os números de 1 a 10

```
int i;
```

```
for ( i=1 ; i <= 10 ; i++ )
```

```
{
```

```
    printf("O número é: %d \n", i );
```

```
}
```

O incremento não precisa ser de um em um.
Ele pode variar qualquer valor.
Pode também diminuir, ou seja, o valor inicial é maior que o valor final (condição de parada)



FOR – Exemplo 2

// Imprime os números pares de 0 a 100

```
int i;
```

```
for ( i=0 ; i <= 100 ; i = i + 2 )
```

```
{
```

```
    printf("O número é: %d \n", i );
```

```
}
```

CUIDADO.

Usar o comando
de atribuição (=).



FOR – Exemplo 3

```
// Fatorial de um número
int i, n, fat;
printf("Informe o número: ");
scanf("%d", &n);
fat = 1;
for ( i=n ; i > 1 ; i--) {
    fat = fat * i;
}
printf("O fatorial é: %d \n", fat );
```



**INSTITUTO
FEDERAL**
Espírito Santo