

# controle de fluxo – repetições



DEPARTAMENTO  
DE INFORMÁTICA  
PUC-RIO

# controle de fluxo – repetições

## tópicos

- while
- for
- do ... while
- break
- Continue

## referências

- Capítulo 5 da apostila
- Capítulo 3 do livro

## como fazer para ...

- “contar” de 1 a 100?
- calcular a média das notas de uma turma de  $n$  alunos, lendo a média de cada aluno e no final apresentar o resultado?
- Permitir que um usuário jogue um jogo várias vezes, até ele pedir para parar de jogar?

# construção while

## Pseudocodigo

```
...  
enquanto (condicao) faça  
    instrução V1  
    instrução V2  
    ...  
    instrução Vn  
fim-enquanto  
...  
fim
```

1) Enquanto **expressão** for verdadeira, executa bloco de comandos

ou

1) Se **expressão** for verdadeira:  
(a) executa bloco de comandos; e  
(b) repete (1)

## C

```
while ( expressao )  
{  
    bloco de comandos  
}
```

## exemplo

```
#include <stdio.h>  
int main (void)  
{  
    int x = 1;  
    while ( x <= 100 )  
    {  
        printf ("%d ", x);  
        ++x;  
    }  
    return 0;  
}
```

# construção while

Modifique o programa abaixo para exibir a soma dos números pares de 1 a 100

```
#include <stdio.h>
int main (void)
{
    int x = 1;
    while ( x <= 100 )
    {
        printf ("%d ", x);
        ++x;
    }
}
```

```
#include <stdio.h>
int main (void)
{
    int x = 1;
    _____

    while ( x <= 100 )
    {
        _____
        _____
        ++x;
    }
    _____

    return 0;
}
```

# construção while

Faça um programa para calcular o fatorial de um número

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

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

# construção while

EX. 03

Escreva um programa que leia o número  $n$  de alunos de uma turma, leia a nota de cada um e exiba a média da turma. Caso  $n \leq 0$ , exiba uma mensagem de erro.

# construção while

```
#include <stdio.h>
int main(void) {
    int n, i=0;
    float nota, soma=0.0F;
    printf("Numero de alunos: ");
    scanf("%d", &n);
    if (n <= 0) {
        printf("Numero de alunos deve ser maior que zero.\n");
        return 1;
    }
    while (i < n) {
        printf("Nota: ");
        scanf("%f", &nota);
        soma += nota;
        ++i;
    }
    printf("A media da turma foi %f\n", soma/n);
    return 0;
}
```

dúvidas?



**DEPARTAMENTO  
DE INFORMÁTICA**  
PUC-RIO

# construção for

```
for ( expr_inicial; expr_teste_laço; expr_atualização )  
{  
    bloco de comandos  
}
```

- 1) avalia **expr\_inicial**
- 2) se **expr\_teste\_laço** for verdadeira:
  - 2.1) executa bloco de comandos
  - 2.2) avalia **expr\_atualização**
  - 2.3) repete a partir de (2)

Construção equivalente usando while:

```
expr_inicial;  
while (expr_teste_laço)  
{  
    bloco de comandos  
    expr_atualização  
}
```

# construção for

EX. 04

Modifique o programa abaixo para utilizar for:

```
#include <stdio.h>
int main (void)
{
    int x = 1;
    while ( x <= 100 )
    {
        printf ("%d ", x);
        ++x;
    }
    return 0;
}
```

```
#include <stdio.h>
int main (void)
{
    _____
    for ( _____; _____; _____ )
    {
        _____
        _____
        _____
    }
    return 0;
}
```

```
for ( expr_inicial; expr_teste_laço; expr_atualização )
{
    bloco de comandos
}
```

# construção for

EX. 05

Utilizando for, escreva um programa para calcular o fatorial de um número não negativo n:

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

dúvidas?



**DEPARTAMENTO  
DE INFORMÁTICA**  
PUC-RIO

# repetição com teste no final: construção for do-while

```
do
{
    bloco de comandos
} while ( expr_teste_laço );
```

- 1) executa bloco de comandos
- 2) se **expressão** for verdadeira,  
repete a partir de (1)

```
int de_novo;
do
{
    jogo();
    printf("Quer jogar novamente (1=sim, 0-nao)? ");
    scanf("%d", &de_novo);
} while (de_novo != 0)
```

# dúvidas?

```
while ( expressao )  
{  
    bloco de comandos  
}
```

- 1) se *expressao* for verdadeira, executa bloco de comandos
- 2) repete a partir de (1)

```
do  
{  
    bloco de comandos  
} while ( expressao );
```

- 1) executa bloco de comandos
- 2) se *expressao* for verdadeira, repete a partir de (1)

```
for ( expr_inicial; expr_teste_laço; expr_atualização )  
{  
    bloco de comandos  
}
```

- 1) avalia *expr\_inicial*
- 2) se *expr\_teste\_laço* for verdadeira:
  - 2.1) executa bloco de comandos
  - 2.2) avalia *expr\_atualização*
  - 2.3) repete a partir de (2)

```
expr_inicial;  
while ( expr_teste_laço )  
{  
    bloco de comandos  
    expr_atualização  
}
```

# laços – interrupções com break e continue

```
#include <stdio.h>
int main (void)
{
    int i;
    for ( i = 0; i < 10; i ++ ) {
        if (i == 5)
            break;
        printf(“%d ”, i);
    }
    -----
    printf (“fim.\n”);
    return 0;
}
```

```
#include <stdio.h>
int main (void)
{
    int i;
    for ( i = 0; i < 10; i ++ ) {
        if (i == 5)
            continue;
        -----
        printf(“%d ”, i);
    }
    printf (“fim.\n”);
    return 0;
}
```

Qual é a saída de cada um desses programas?

dúvidas?



**DEPARTAMENTO  
DE INFORMÁTICA**  
PUC-RIO