

Laços aninhados

Aula 07

Marcos Silvano Almeida

marcossilvano@professores.utfpr.edu.br

Departamento de Computação

UTFPR Campo Mourão

BCC31A

Roteiro

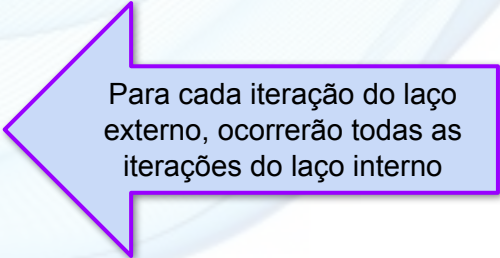
- Laços aninhados
- Escopos de variáveis
- Adendos:
 - *Tabela Unicode*
 - *Como imprimir colorido ⇒ Moodle*

Laços Aninhados

Laços Aninhados

- Dentro de um bloco de comandos { } podemos colocar
 - Repetição de comandos
 - Comandos de seleção: IF, IF-ELSE, SWITCH
 - Comandos de repetição: WHILE, FOR, DO-WHILE
- Logo, é possível colocar um LOOP dentro de outro LOOP

```
while(condição) {    // LOOP EXTERNO
    linha1;
    while(condição) {    // LOOP INTERNO
        linha2;
        linha3;
        ...
    }
    linha4;
    ...
}
```



Para cada iteração do laço externo, ocorrerão todas as iterações do laço interno

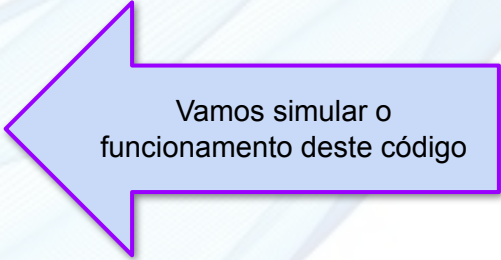
Laços Aninhados: exemplo

- Imprimindo a cada iteração dos laços:

```
#include <stdio.h>

int main() {
    for (int i = 0; i < 3; i++) {
        printf("Outer loop: %d\n", i);

        for (int j = 0; j < 4; j++) {
            printf("    Inner loop: %d\n", j);
        }
    }
    return 0;
}
```



Vamos simular o
funcionamento deste código

Laços Aninhados: exemplo 2

- Qual o valor impresso ao final?

```
#include <stdio.h>
```

```
int main() {
```

```
    int count = 0;
```

```
    for (int i = 0; i < 2; i++) {
```

```
        for (int j = 0; j < 4; i++) {
```

```
            count++;
```

```
        }
```

```
    }
```

```
    printf("Count: %d\n", count);
```

```
    return 0;
```

```
}
```

Laços Aninhados: exemplo 3 + Unicode

- Podemos escrever o padrão Unicode na linguagem C

```
#include <stdio.h>
// TABELA UNICODE:
// https://en.wikipedia.org/wiki/List_of_Unicode_characters
```

```
int main() {
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 5; j++) {
            printf("😊"); // \U0001F603
        }
        printf("\n");
    }
    return 0;
}
```

SAÍDA: 4x5



Escopos de variáveis

Escopos de variáveis

- Um escopo { } define a extensão de código em que a variável é acessível
 - Variáveis podem ser acessadas somente no escopo em que foram definidas ou em escopos descendentes, a partir de sua declaração

```
int main() {  
    int num = 0;                                // acessível em todo o escopo de main()  
    for (int i = 0; i < 10; i++) {                // i existe apenas dentro do escopo do for  
        printf("> ");  
        scanf("%d", &num);  
        if (num < 0) {                            // acessando i em escopo descendente  
            printf("*AVISO* Entrada cancelada no passo %d! \n", i+1);  
            break;  
        }  
    }  
    printf("Ultimo passo: %d\n", i);              // i não é acessível fora do escopo do FOR  
    printf("Ultimo numero informado: %d\n", num);  
    return 0;  
}
```

Escopos de variáveis

- O escopo global permite o acesso universal à variável, ao custo de mantê-la na memória durante a execução inteira do programa (evitar)

```
int globalNum = 0;                                // variável GLOBAL (EVITE!)
int main() {
    int rep;                                       // variável do escopo da main()
    printf("\nLOOPS:\n> ");
    scanf("%d", &rep);
    for (int i = 0; i < rep; i++) {               // i existe no escopo do FOR
        if (i % 2 == 0) {                         // i é par?
            int multi = i * 10;                   // multi existe no escopo do IF
            printf("[%d] ", multi);
            globalNum += multi;                   // acessando variável global
        } else {
            printf("%d ", i);
        }
    }
}
```

Colorindo o terminal com
Escape Sequences

Cores :: sequências de saída

FG = Foreground -> cor do caractere

BG = Background -> cor do fundo

```
USO: printf("\033[%dm", CODIGO_DA_COR);
```

>> CORES PARA O CARACTERE_____

BLACK	30	RED	31
GREEN	32	YELLOW	33
BLUE	34	MAGENTA	35
CYAN	36	WHITE	37

>> CORES PARA O FUNDO_____

BLACK	40	RED	41
GREEN	42	YELLOW	43
BLUE	44	MAGENTA	45
CYAN	46	WHITE	47

RESET 0 // retorna a cor original

Escape Sequences são sequências de controle que não imprimem caracteres no terminal, mas realizam ações.

Na linha C, iniciam com \

Exemplo:

\033[31m => cor do texto para vermelho

Cores :: macros em arquivo utfcolors.h

```
#define COLOR_FG_BLACK    30
#define COLOR_FG_RED      31
#define COLOR_FG_GREEN    32
#define COLOR_FG_YELLOW   33
#define COLOR_FG_BLUE     34
#define COLOR_FG_MAGENTA  35
#define COLOR_FG_CYAN     36
#define COLOR_FG_WHITE    37
```

```
#define COLOR_BG_BLACK    40
#define COLOR_BG_RED      41
#define COLOR_BG_GREEN    42
#define COLOR_BG_YELLOW   43
#define COLOR_BG_BLUE     44
#define COLOR_BG_MAGENTA  45
#define COLOR_BG_CYAN     46
#define COLOR_BG_WHITE    47
```

```
#define COLOR_RESET 0
#define SET_COLOR(color) printf("\033[%dm", color);
```

Na linguagem C, utilizamos MACROS para definir símbolos, a serem substituídos no código por seus conteúdos.

Formato:
#define SÍMBOLO valor

Exemplo:
#define MAX_VALUE 50

Biblioteca de cores:

Podemos escrever este código em um arquivo (ex: utfcolors.h) e incluí-lo em nosso código com

#include "utfcolors.h"

Referências

- Algoritmos e Programação
 - Marcela Gonçalves dos Santos
 - Disponível pelo Moodle
- Estruturas de Dados, Waldemar Celes e José Lucas Rangel
 - PUC-RIO - Curso de Engenharia
 - Disponível pelo Moodle
- Linguagem C, Silvio do Lago Pereira
 - USP - Instituto de Matemática e Estatística
 - Disponível pelo Moodle
- Curso Interativo da Linguagem C
 - <https://www.tutorialspoint.com/cprogramming>