



## **Aula 15: Comandos de Repetição**

### **Introdução a Programação**

---

**Túlio Toffolo & Puca Huachi**  
<http://www.toffolo.com.br>

BCC201 – 2019/2  
Departamento de Computação – UFOP

## Aula Anterior

- Prova 01
- Correção da Prova 01
- Trabalho Prático

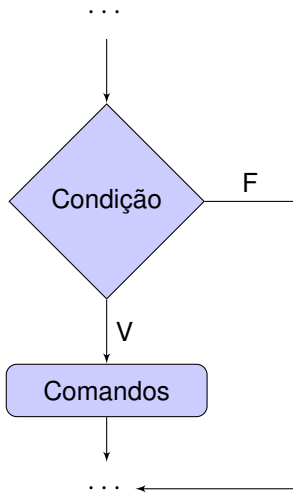
# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`
- 4 Exercícios
- 5 Próximas Aulas

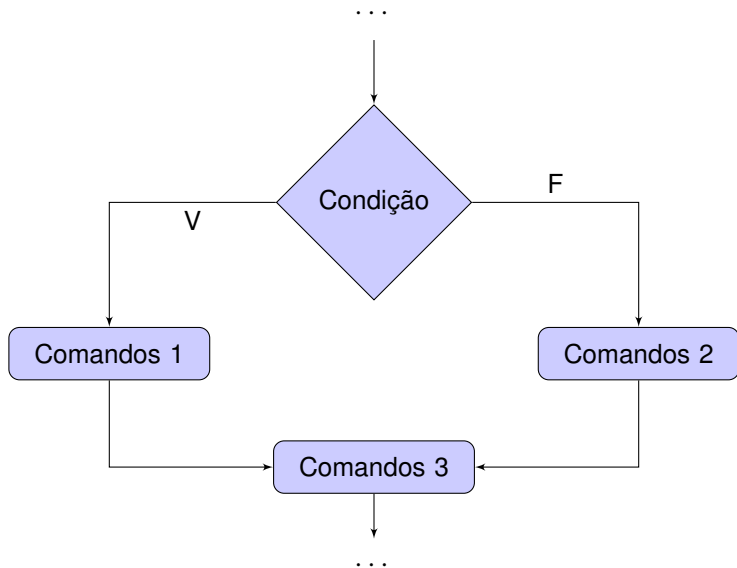
# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`
- 4 Exercícios
- 5 Próximas Aulas

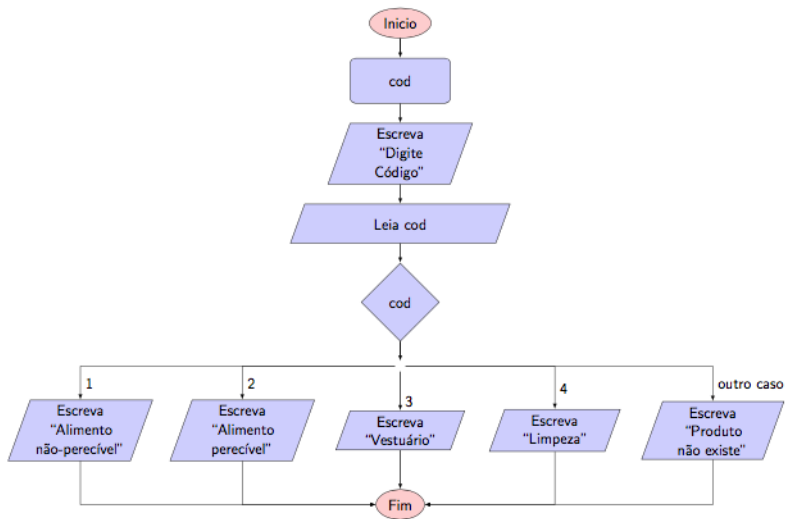
## Tomada de decisão



## Tomada de decisão



# Seleção



## Exemplo

Escreva um programa que calcula os quadrados e cubos dos números inteiros de 0 a 10. Use tabulação para imprimir a seguinte tabela de valores bem formatada:

número	quadrado	cubo
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000



## Exemplo

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      printf("nro \t quad. \t cubo\n");
7      printf("%d \t %.0lf \t %.0lf\n", 0, pow(0, 2), pow(0, 3));
8      printf("%d \t %.0lf \t %.0lf\n", 1, pow(1, 2), pow(1, 3));
9      printf("%d \t %.0lf \t %.0lf\n", 2, pow(2, 2), pow(2, 3));
10     printf("%d \t %.0lf \t %.0lf\n", 3, pow(3, 2), pow(3, 3));
11     printf("%d \t %.0lf \t %.0lf\n", 4, pow(4, 2), pow(4, 3));
12     printf("%d \t %.0lf \t %.0lf\n", 5, pow(5, 2), pow(5, 3));
13     printf("%d \t %.0lf \t %.0lf\n", 6, pow(6, 2), pow(6, 3));
14     printf("%d \t %.0lf \t %.0lf\n", 7, pow(7, 2), pow(7, 3));
15     printf("%d \t %.0lf \t %.0lf\n", 8, pow(8, 2), pow(8, 3));
16     printf("%d \t %.0lf \t %.0lf\n", 9, pow(9, 2), pow(9, 3));
17     printf("%d \t %.0lf \t %.0lf\n", 10, pow(10, 2), pow(10, 3));
18
19     return 0;
20 }
```

## Exemplo

- **E se fosse para calcular até 1000?**

# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`
- 4 Exercícios
- 5 Próximas Aulas

## Laços ou Repetições (*loop*)

- Laços são comandos usados sempre que uma ou mais instruções devam ser repetidas enquanto um certa condição estiver sendo satisfeita
- Laços em C/C++
  - `while`
  - `do - while`
  - `for`
- Programação Estruturada: **sequência**, **desvios** e **repetição**

## Laços ou Repetições (*loop*)

- Em um laço **controlado logicamente**, os comandos (corpo do laço) são repetidos enquanto uma **expressão lógica for verdadeira**
  - `while`
  - `do - while`
- Em um laço **controlado por contador**, os comandos (corpo do laço) são repetidos um **determinado número de vezes**
  - `for`
- Denomina-se **iteração** a repetição de um conjunto de comandos: cada execução do corpo do laço, juntamente com a condição de terminação do laço, é uma iteração

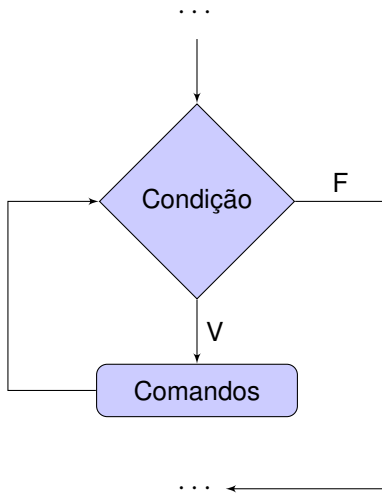
# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`**
- 4 Exercícios
- 5 Próximas Aulas

## Comando `while`

- O comando `while` consiste na palavra chave, `while`, seguida de uma expressão de teste entre parênteses.
- Se a expressão de teste for verdadeira, o corpo do laço `while` é executado e a expressão de teste é avaliada novamente.
- Este ciclo de teste e execução é repetido enquanto a expressão do teste for verdadeira. Quando a expressão se tornar falsa, o laço termina e a execução continua na linha seguinte ao laço.

## Comando `while`





## Comando `while`

- Sintaxe:

```
1 while (<expressão>
2 {
3     <comando_1>;
4     ...
5     <comando_n>;
6 }
```

- As linhas 2–6 representam o corpo do laço.

## Exemplo 1

Escreva um programa que calcula os quadrados e cubos dos números inteiros de 0 a 100. Formate a tabela assim como no exemplo anterior:

número	quadrado	cubo
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
...	...	...
99	9801	970299
100	10000	1000000

## Exemplo 1

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int i;
7      printf("nro \t quad. \t cubo\n");
8
9      i = 0; // inicializa a variável
10     while (i <= 100) {
11         printf("%d \t %.0lf \t %.0lf\n", i, pow(i, 2), pow(i, 3));
12         i++; // equivalente: i = i + 1
13     }
14
15     return 0;
16 }
```

## Importante:

- Não pode esquecer de inicializar o contador (linha 9)
  - Se não for inicializado pode começar com qualquer valor, ou seja, **lixo de memória**.
- Não pode esquecer de incrementar o contador (linha 14)
  - se o contador não for incrementado, o laço nunca terá fim, ou seja, teremos um **loop infinito**.

## Exemplo 2

Faça um programa que leia as notas da primeira prova dos  $n$  alunos de BCC201 e calcule e imprima a média das notas.

## Exemplo 2

```
1 #include <stdio.h>
2
3 int main()
4 {
5     double nota, media, soma;
6     int alunos, i;
7
8     printf("Digite o nro de alunos: ");
9     scanf("%d", &alunos);
10
11     i = 1;
12     soma = 0;
13     while (i <= alunos) {
14         printf("Digite a nota do aluno %d: ", i);
15         scanf("%lf", &nota);
16         soma += nota;
17         i++;
18     }
19     media = soma / alunos;
20     printf("A média das notas é: %.2lf\n", media);
21     return 0;
22 }
```

## Exemplo 3

Faça um programa que leia as notas da primeira prova de BCC201 e calcule e imprima a média das notas dos  $n$  alunos da disciplina.

**O programa deve verificar se nota digitada está correta, ou seja, garantir que  $0 \leq nota \leq 10$ .**

```
1 int main()
2 {
3     double nota, media, soma;
4     int alunos, i;
5
6     printf("Digite o nro de alunos: ");
7     scanf("%d", &alunos);
8
9     i = 1;
10    soma = 0;
11    while (i <= alunos) {
12        printf("Digite a nota do aluno %d: ", i);
13        scanf("%lf", &nota);
14        if ((nota < 0) || (nota > 10)) {
15            printf("Nota inválida! Digite novamente: ");
16        }
17        else {
18            soma += nota;
19            i++;
20        }
21    }
22    media = soma / alunos;
23    printf("A média das notas é: %.2lf\n", media);
24    return 0;
25 }
```



## Exemplo 4

Faça um programa que leia as notas da primeira prova de BCC201 e calcule e imprima a média das notas.

Considere que o número de alunos é desconhecido. Utilize como critério de parada do programa uma nota negativa.

## Exemplo 4

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int alunos = 0;
6     double nota, soma = 0;
7
8     printf("Digite um nro negativo para sair ou a nota do aluno: ");
9     scanf("%lf", &nota);
10    while (nota >= 0) {
11        soma += nota;
12        alunos++;
13        printf("Digite um nro negativo para sair ou a nota do aluno: ");
14        scanf("%lf", &nota);
15    }
16    double media = soma / alunos;
17    printf("A média das notas é: %.2lf\n", soma);
18    return 0;
19 }
```

## Exemplo 4 (Alternativa 2)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int alunos = 0;
6      double nota = 0, media = 0;
7
8      while (nota >= 0) {
9          printf("Digite um nro negativo para sair ou a nota do aluno: ");
10         scanf("%lf", &nota);
11
12         if (nota >= 0) {
13             soma += nota;
14             alunos++;
15         }
16     }
17     double media = soma / alunos; // equivale a media = soma / alunos;
18     printf("A média das notas é: %.2lf\n", media);
19     return 0;
20 }
```

## Exemplo 5

Escreva um programa que imprima o valor de  $S$ , calculado a partir do somatório a seguir para um dado valor de  $n$  (este valor será digitado pelo usuário e deve ser tal que  $n > 0$ ).

$$S = \sum_{i=0}^{n-1} \frac{1}{2^i}$$

Exemplo de execução:

```
1 Digite o valor de n (inteiro maior que zero): 5
2 Resultado: 1.937500
```

## Exemplo 5

```
1 #include <math.h>
2 #include <stdio.h>
3
4 int main()
5 {
6     int n;
7     printf("Digite o valor de n (inteiro maior que zero): ");
8     scanf("%d", &n);
9
10    int i = 0;
11    double s = 0;
12    while (i < n) { // equivale a: while (i <= n-1) {
13        s = s + 1/pow(2,i);
14        i++; // equivale a: i = i + 1;
15    }
16
17    printf("Resultado: %lf\n", s);
18    return 0;
19 }
```

## Exemplo 6

Escreva um programa - uma **calculadora** - que permite ao usuário efetuar as 4 operações (+, -, \*, /) básicas diversas vezes.

- O programa solicita a operação e dois operandos para, em seguida, retornar o resultado.
- O programa termina quando a operação digitada for “#”. Para qualquer outro símbolo digitado diferente das 4 operações e do critério de parada, o programa deve informar que a operação é inválida.

```
1 int main()
2 {
3     double x, y, res;
4     char op;
5     scanf("%c", &op);
6
7     while (op != '#') {
8         scanf("%lf %lf", &x, &y);
9         switch (op) {
10            case '+':
11                printf("%lf + %lf = %lf\n", x, y, x+y);
12                break;
13            case '-':
14                printf("%lf - %lf = %lf\n", x, y, x-y);
15                break;
16            case '*':
17                printf("%lf * %lf = %lf\n", x, y, x*y);
18                break;
19            case '/':
20                printf("%lf / %lf = %lf\n", x, y, x/y);
21                break;
22            default:
23                printf("Operação inválida!\n");
24                break;
25        }
26        scanf("\n%c", &op);
27    }
28    return 0;
29 }
```

# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`
- 4 Exercícios**
- 5 Próximas Aulas



## Exercício 1

A média harmônica é dada pela seguinte equação:

$$H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}, \text{ ou seja: } H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

Faça um programa em C para calcular a média harmônica de  $n$  valores digitados pelo usuário. Saída:

```
1  Cálculo da Média Harmônica.  
2  Quanto números você quer digitar? 4  
3  
4  Número 1: 2  
5  Número 2: 4  
6  Número 3: 6  
7  Número 4: 8  
8  
9  Média Harmônica: 3.84
```

# Aula de Hoje

- 1 Motivação
- 2 Laços
- 3 Comando `while`
- 4 Exercícios
- 5 Próximas Aulas

## Próximas Aulas

- Comandos de repetição (Parte 2)
  - `do - while`
- Comandos de repetição (Parte 3)
  - `for`



Perguntas?