



UNIAVAN - Centro Universitário Avantis  
Curso: Sistemas de Informação  
Disciplina: Algoritmos I

# Programação em C

## Estruturas/Laço de Repetição

Prof. Luiz Fernando M. Arruda, Me. Eng.

- 1 Estruturas ou laços de repetição
- 2 Estrutura de Repetição - While
- 3 Estrutura de Repetição - do While
- 4 Estrutura de Repetição - For
- 5 Vamos Praticar!?

As estruturas ou laços de repetição são artifícios/métodos muito utilizados em programação para efetuar uma determinada atividade/processo/função por diversas vezes utilizando a mesma estrutura lógica, porém de forma automatizada, ou seja, o programador desenvolve uma única vez e pede para que o sistema repita a operação por " $n$ " vezes.

Existem 3 estruturas de repetição.

- *while* - Enquanto
- *do while* - Faça enquanto
- *for* - Para

A principal diferença entre estes modelos é que enquanto o *while* aguarda um evento específico, o *for* possui um variável de controle incremental.

## Curiosidade

Após compreendido o conceito de um laço de repetição, este recurso pode ser utilizado em qualquer linguagem de programação.





Na implementação de uma estrutura *While*, parte-se do princípio que a variável de controle utilizada aguarda um evento. Por exemplo, pode-se repetir até que o usuário tecele uma tecla específica, pode-se repetir até que uma determinada temperatura seja alcançada, ou qualquer evento que não necessariamente seja incremental, mas que oscile tempo, ou aguarde um evento por parte do usuário.

Este recurso é executado enquanto a lógica do bloco for verdadeira.

---

```
while (condição) {  
    // Código a ser executado se a condição for verdadeira  
}
```

---



```
#include <stdio.h>

int main (void){
    int num = 1;
    printf("Exibir os 10 primeiros números:\n");
    while (num <= 10){
        printf("%d\n", num);
        num = num + 1;
    }
    return 0;
}
```



```
#include <stdio.h>
int main (void){
    int num = 1;
    int valor = 0;
    int acumulador = 0;
    float media = 0.0;
    printf("Calcule a média dos 10 primeiros números digitados.\n");
    printf("Digite os números em sequência\n");
    while (num <= 10){
        scanf("%d", &valor);
        acumulador = acumulador + valor;
        num = num + 1;
    }
    media = acumulador / 10;
    printf("A média é %.2f",media);
    return 0;
}
```





1. Faça um algoritmo que escreva na tela os números de zero até 100.
2. Faça um algoritmo que escreva na tela os números de zero até um número digitado pelo usuário.
3. Faça um algoritmo que escreva na tela os números inteiros existentes entre dois números digitados pelo usuário.
4. Faça um algoritmo que escreva a tabuada de um número inteiro digitado pelo usuário, indo de 1 a 10.
5. Escrever um algoritmo que e escreva os números pares inteiros existentes entre 100 e 200;
6. Escrever um programa de computador que leia 10 números inteiros, e ao final, apresente a soma de todos os números lidos;
7. Faça um programa que some os valores fornecidos pelo usuário até que o usuário entre com 0 para interromper o programa.

A estrutura de repetição *do While*, é semelhante ao recurso de *While*, havendo apenas uma única diferença. A estrutura *While* faz a verificação da condição no início do processo, já a *do While*, primeiro executa e depois verifica se continua ou não a executar a estrutura.

```
do{  
    // Código a ser repetido se a condição for verdadeira  
}while (condição);
```

### Cuidado

A lógica do bloco de validação muda conforme a linguagem de programação.



```
#include <stdio.h>

int main (void){
    int num = 1;
    printf("Exibir os 10 primeiros números:\n");
    do{
        printf("%d\n", num);
        num = num + 1;
    } while (num <= 10);
    return 0;
}
```



```
#include <stdio.h>
int main (void){
    int num = 1;
    int valor = 0;
    int acumulador = 0;
    float media = 0.0;
    printf("Calcule a média dos 10 primeiros números digitados.\n");
    printf("Digite os números em sequência\n");
    do{
        scanf("%d", &valor);
        acumulador = acumulador + valor;
        num = num + 1;
    } while (num <= 10);
    media = acumulador / 10;
    printf("A média é %.2f",media);
    return 0;
}
```



8. Escreva um algoritmo que calcule a média dos números digitados pelo usuário, se eles forem pares. Termine a leitura se o usuário digitar zero (0);
9. Escreva um algoritmo que leia valores inteiros e encontre o maior e o menor deles. Termine a leitura se o usuário digitar zero (0);
10. Escreva um programa que lê o sexo de uma pessoa. O sexo deverá ser com o tipo de dado caractere e o programa deverá aceitar apenas os valores "M" ou "F".
11. Escreva um programa que leia dois valores reais. Ambos valores deverão ser lidos até que o usuário digite um número no intervalo de 1 a 100. Apresentar a soma dos dois valores lidos.
12. Escrever um algoritmo que gera e escreve os números ímpares entre 100 e 200;
13. Em uma turma de programação há 15 alunos matriculados. Cada aluno possui 3 notas. Crie um programa que calcule a média da **turma** para cada uma das notas;
14. Faça um programa que some os valores fornecidos pelo usuário até que o usuário entre com 0 para interromper o programa.

A estrutura de repetição *For* é sem sombra de dúvida a mais utilizada em programação. Ela para executa uma instrução em um número de repetições previamente definido.

Ela possui três parâmetros:

- Inicialização da variável de controle;
- Condição para repetição;
- Variável de auto incremento;

---

```
for( inicialização ; condição ; incremento){  
    // Código a ser repetido se a condição for verdadeira  
};
```

---



```
#include <stdio.h>

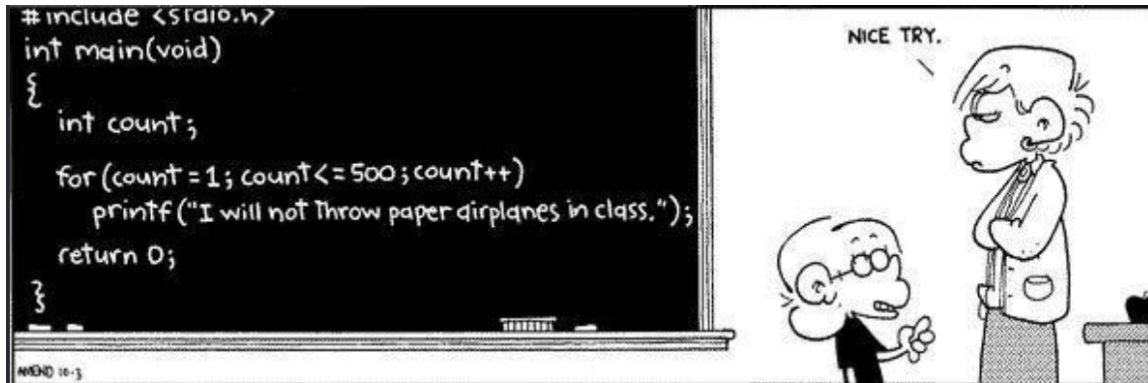
int main (void){
    int num = 0;
    printf("Exibir os 10 primeiros números:\n");
    for (num = 1; num <= 10; num++)
    {
        printf("%d\n", num);
    }
    return 0;
}
```



```
#include <stdio.h>
int main (void){
    int num = 1;
    int valor = 0;
    int acumulador = 0;
    float media = 0.0;
    printf("Calcule a média dos 10 primeiros números digitados.\n");
    printf("Digite os números em sequência\n");
    for(num = 1; num <= 10; num++){
        scanf("%d", &valor);
        acumulador = acumulador + valor;
    }
    media = acumulador / 10;
    printf("A média é %.2f",media);
    return 0;
}
```







15. Escreva um algoritmo que faz a leitura de 5 números inteiros e apresente o menor e o maior deles;
16. Escreva um algoritmo que recebe 8 números inteiros e efetue a soma somente dos números ímpares;
17. Escreva um algoritmo que recebe 8 números inteiros e efetue a soma somente dos números pares;
18. Faça um algoritmo que leia a idade, altura e peso de 10 pessoas e mostre: a. o número de pessoas com idade maior de 30 anos. b. a média das alturas das pessoas entre 20 e 30 anos. c. a porcentagem de pessoas acima de 60 Kg.



# moodle

<https://moodle.electronicssystem.com.br>

ARAUJO, Everton. **Algoritmos: Fundamentos e prática**. [S.I.]: Visual Book, 2007.

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java**. [S.I.]: São Paulo: Pearson Education do Brasil, 2012.

DEITEL, Paul; DEITEL, Harvey; STEINBUHLER, K. **C Como Programar**. 6ª Edição. [S.I.]: Pearson Prentice Hall, 2011.

KAWANO, Wilson. **Migrando C para C++: guia prático de programação**. [S.I.]: Ciência Moderna, 2010.

MANANO, João Augusto; NAVARRO, Garcia. **Programação de computadores com C/C++**. [S.I.]: Érica, 2014.

MEDINA, Marco; FERTIG, Cristina. **Algoritmos e programação: teoria e prática**. [S.I.]: Novatec, 2006.

OLIVEIRA, Alvaro Borges de; BORATTI, Isaias Camilo. **Introdução à Programação: Algoritmos**. [S.l.]: Florianópolis: Bookstore, 1999.

ZIVIANI, Nivio. **Projetos de Algoritmos com implementações em Pascal e C**. [S.l.]: Pioneira Thomson Learning, 2004.