



Universidade Positivo

# Algoritmos de Programação

Aula 11

Prof.<sup>a</sup> Mariane Cassenote

[mariane.cassenote@up.edu.br](mailto:mariane.cassenote@up.edu.br)

# Conteúdo



- Estruturas de Repetição
- Continuidade da introdução à programação em linguagem C

Ao final do componente curricular será possível utilizar estruturas de repetição simples e aninhadas para resolver problemas algorítmicos.

# Estruturas de Repetição

*// Leia um número do teclado e o imprima 10 vezes*

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

```
int main(void) {
```

```
    int num;
```

```
    printf("Digite um numero inteiro: ");
```

```
    scanf("%d", &num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    printf("%d", num);
```

```
    return 0;
```

```
}
```

Digite um numero inteiro: 3

3  
3  
3  
3  
3  
3  
3  
3  
3  
3  
3

# Estruturas de Repetição



- Até este momento, estudamos algoritmos que executam comandos **em sequência**, um após outro, uma única vez ou até mesmo nenhuma vez (em alguns casos de uso de estruturas condicionais / de decisão)
- No entanto, existem situações bastante comuns na computação em que precisamos executar blocos de comandos **repetidas vezes**
- Uma estrutura de repetição (**laço / loop**) permite executar diversas vezes um bloco de comandos diante da avaliação de uma condição (expressão lógica)

# Estruturas de Repetição



- O número de repetições pode ser previamente conhecido ou estar associado à ocorrência de uma condição específica que é testada no decorrer do processamento
- Exemplos:
  - cálculo da média da turma
  - atualização de registros
  - operação de caixa de supermercado

# Estruturas de Repetição



Existem três tipos mais comuns de estruturas de repetição:

- Com **teste no início**: **WHILE** (enquanto)
- Com **teste no final**: **DO... WHILE** (faça-enquanto)
- Guiada por **contador**: **FOR** (para)

# Estruturas de Repetição



Para se criar uma estrutura de repetição, além do bloco de comandos a serem executados, devem / podem ser definidos:

- Uma forma de indicar o **retorno** a um determinado ponto anterior do código para repetição
- Um recurso para registrar o número de vezes que a repetição foi realizada, como um **contador** (opcional)
- Uma **condição** a ser testada que permita determinar se a repetição deve ser executada novamente ou não



# Estruturas de Repetição

`while`

`enquanto`



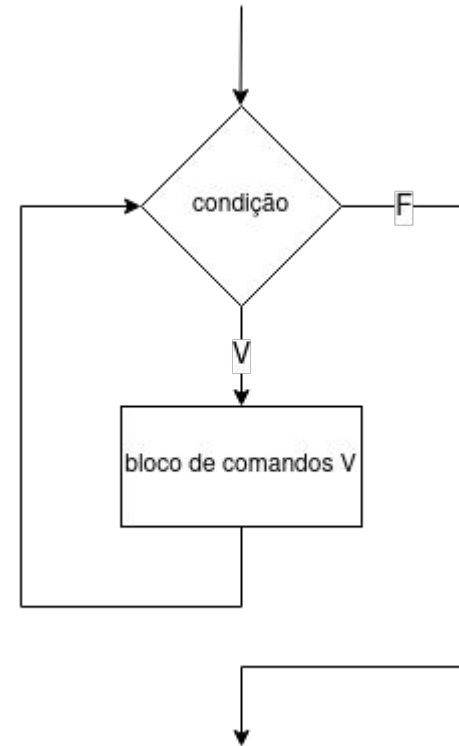
# Estruturas de Repetição – enquanto...faça / while

Em pseudocódigo:

```
enquanto (<condição>) {  
    <bloco de comandos para condição verdadeira>  
}
```

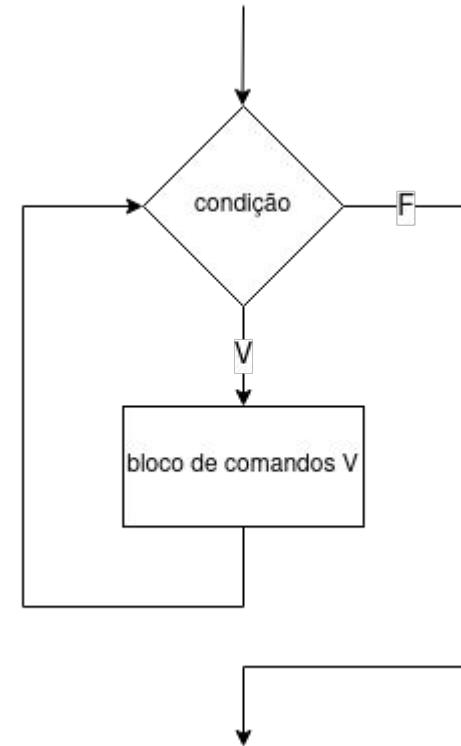
Em linguagem C:

```
while (<condição>) {  
    <bloco de comandos para condição verdadeira>  
}
```



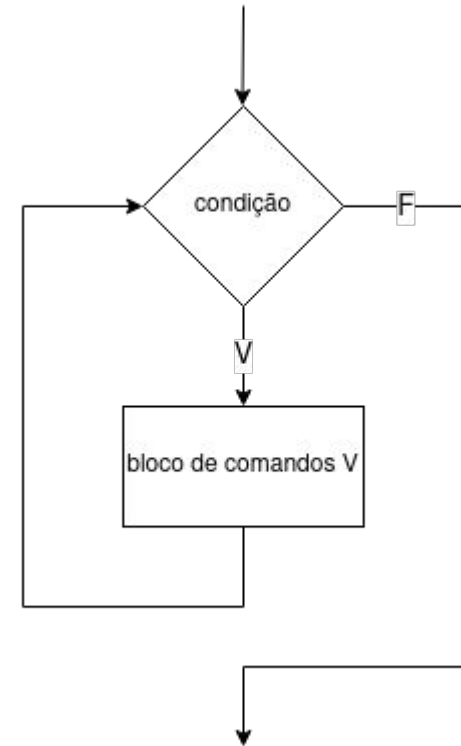
# Estruturas de Repetição – enquanto...faça / while

- Primeiro testa a **condição**, depois executa o bloco de comandos
- A execução de um bloco de comandos é repetida **enquanto** a avaliação da condição tiver resultado **VERDADEIRO**
- Quando a condição passa a ser **FALSA**, a repetição é **encerrada**



# Estruturas de Repetição – enquanto...faça / while

- O bloco de comandos **nunca será executado** se a condição for **FALSA** logo de início
- Se a condição for VERDADEIRA, é necessário que alguma **alteração** ocorra durante a execução do bloco de comandos para que em algum momento a condição seja avaliada como FALSA. Caso contrário, se tem um **laço (loop) infinito**



**Cuidado com os loops infinitos!**

# Estruturas de Repetição - Contador



- Relacionado a repetições por **contagem**
- Serve para contabilizar o **número** de repetições ou para determinar seu **encerramento**
- Utilizar como uma **variável do tipo inteiro**
- Exemplo da catraca

```
contador = 0;           // inicialização fora da estrutura de repetição  
contador = contador + 1; // incremento em uma unidade a cada repetição
```

# Estruturas de Repetição – enquanto...faça / while

```
#include <stdio.h>

int main() {
    int contador;

    contador = 1;

    while (contador <= 5) {
        printf("Eh o carro do sonho, freguesia %d \n", contador);
        contador = contador + 1;
    }

    return 0;
}
```

```
Eh o carro do sonho, freguesia 1
Eh o carro do sonho, freguesia 2
Eh o carro do sonho, freguesia 3
Eh o carro do sonho, freguesia 4
Eh o carro do sonho, freguesia 5
```

# Estruturas de Repetição – enquanto...faça / while

```
#include <stdio.h>

int main() {
    int contador;

    contador = 1;

    while (contador <= 5) {
        printf("Eh o carro do sonho, freguesia %d\n", contador);
        contador = contador + 1;
    }

    return 0;
}
```

O contador é inicializado com 1 (um) e é incrementado em uma unidade a cada repetição do laço enquanto seu valor for menor ou igual a 5.

Quando ele atingir o valor 6, o laço é encerrado.

```
ro do sonho, freguesia 1
Eh o carro do sonho, freguesia 2
Eh o carro do sonho, freguesia 3
Eh o carro do sonho, freguesia 4
Eh o carro do sonho, freguesia 5
```

# Estruturas de Repetição – enquanto...faça / while

```
#include <stdio.h>

int main() {
    int contador;

    contador = 1;

    while (contador <= 5) {
        printf("Eh o carro do sonho, freguesia %d\n", contador);
        contador = contador + 1;
    }

    return 0;
}
```

Mas programadores não costumam  
começar a contar em 0 (zero)?  
E se eu quiser que meu contador  
inicie em 0 (zero)?  
Basta ajustar a condição

```
Eh o carro do sonho, freguesia 1
Eh o carro do sonho, freguesia 2
Eh o carro do sonho, freguesia 3
Eh o carro do sonho, freguesia 4
Eh o carro do sonho, freguesia 5
```

# Estruturas de Repetição – enquanto...faça / while

```
#include <stdio.h>

int main() {
    int contador;

    contador = 0;

    while (contador < 5) {
        printf("Eh o carro do sonho, freguesia %d \n", contador);
        contador = contador + 1;
    }

    return 0;
}
```

```
Eh o carro do sonho, freguesia 0
Eh o carro do sonho, freguesia 1
Eh o carro do sonho, freguesia 2
Eh o carro do sonho, freguesia 3
Eh o carro do sonho, freguesia 4
```



# Estruturas de Repetição – enquanto...faça / while

```
programa {
    funcao inicio() {

        cadeia frase
        inteiro contador

        escreva("Digite uma frase: ")
        leia(frase)

        contador = 1

        enquanto (contador <= 5) {
            escreva(frase + " " + contador + "\n")
            contador = contador + 1
        }

    }
}
```

```
Digite uma frase: Eh o carro do sonho, freguesia!
Eh o carro do sonho, freguesia! 1
Eh o carro do sonho, freguesia! 2
Eh o carro do sonho, freguesia! 3
Eh o carro do sonho, freguesia! 4
Eh o carro do sonho, freguesia! 5

Programa finalizado.
```

# Estruturas de Repetição – enquanto...faça / while

```
programa {
    funcao inicio() {

        cadeia frase
        inteiro contador

        escreva("Digite uma frase: ")
        leia(frase)

        contador = 0

        enquanto (contador < 5) {
            escreva(frase + " " + contador + "\n")
            contador = contador + 1
        }

    }
}
```

```
Digite uma frase: Eh o carro do sonho, freguesia!
Eh o carro do sonho, freguesia! 0
Eh o carro do sonho, freguesia! 1
Eh o carro do sonho, freguesia! 2
Eh o carro do sonho, freguesia! 3
Eh o carro do sonho, freguesia! 4
```

```
Programa finalizado.
```

# Estruturas de Repetição – enquanto...faça / while



Leia cinco números do teclado e os imprima multiplicados por 3.

# Estruturas de Repetição – enquanto...faça / while

*// Leia cinco números do teclado e os imprima multiplicados por 3*

```
#include <stdio.h>
int main(void) {

    int contador, numero, resultado;
    contador = 0;

    while (contador < 5) {
        printf("Digite um numero inteiro: ");
        scanf("%i", &numero);

        resultado = numero * 3;
        printf("Resultado: %i \n", resultado);

        contador = contador + 1;
    }
    return 0;
}
```

```

Digite um numero inteiro: 2
Resultado: 6
Digite um numero inteiro: 3
Resultado: 9
Digite um numero inteiro: 4
Resultado: 12
Digite um numero inteiro: 5
Resultado: 15
Digite um numero inteiro: 6
Resultado: 18
```



# Estruturas de Repetição – enquanto...faça / while

## // EM PSEUDOCÓDIGO

```
programa {  
    funcao inicio() {  
  
        inteiro contador, numero, resultado  
        contador = 0  
  
        enquanto (contador < 5) {  
            escreva("Digite um numero inteiro: ")  
            leia(numero)  
  
            resultado = numero * 3  
            escreva("Resultado: ", resultado, "\n")  
  
            contador = contador + 1  
        }  
    }  
}
```

## // EM LINGUAGEM C

```
#include <stdio.h>  
  
int main(void) {  
  
    int contador, numero, resultado;  
    contador = 0;  
  
    while (contador < 5) {  
        printf("Digite um numero inteiro: ");  
        scanf("%i", &numero);  
  
        resultado = numero * 3;  
        printf("Resultado: %i \n", resultado);  
  
        contador = contador + 1;  
    }  
    return 0;  
}
```

# Estruturas de Repetição – enquanto...faça / while



E se não soubermos quantas vezes o  
bloco de comandos deverá ser repetido?

# Estruturas de Repetição – enquanto...faça / while



Elaborar um programa que efetue a leitura de um valor numérico inteiro qualquer do teclado. Em seguida, multiplique esse valor por 3 e apresente o resultado.

Após apresentar o resultado, o programa deve perguntar ao usuário se ele deseja realizar um novo cálculo. Se a resposta for sim, deve executar novamente.

# Estruturas de Repetição – enquanto...faça / while



Elaborar um programa que

Após apresentar o res

c

Nesse caso, o programa será executado  
conforme **a vontade do usuário**, então não  
há a possibilidade de prever quantas  
repetições irão ocorrer  
**(programa interativo)**

er do teclado. Em seguida,

eseja realizar um novo  
e.



# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado;
```

```
    char continua;
```

```
    continua = 'S'; // aspas simples em char
```

```
    while (continua == 'S') {
```

```
        printf("Digite um numero inteiro: \n");
```

```
        scanf("%d", &numero);
```

```
        resultado = numero * 3;
```

```
        printf("Resultado: %d \n", resultado);
```

```
        printf("Deseja continuar? Digite S para sim ou N para nao: \n");
```

```
        scanf(" %c", &continua); // colocar espaço antes do %
```

```
        printf("----- \n");
```

```
    }
```

```
    return 0;
```

```
}
```

```
Digite um numero inteiro: 2
```

```
Resultado: 6
```

```
Deseja continuar? Digite S para sim ou N para nao: S
```

```
-----
```

```
Digite um numero inteiro: 3
```

```
Resultado: 9
```

```
Deseja continuar? Digite S para sim ou N para nao: S
```

```
-----
```

```
Digite um numero inteiro: 5
```

```
Resultado: 15
```

```
Deseja continuar? Digite S para sim ou N para nao: N
```

```
-----
```

```
> 
```

# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado;
```

```
    char continua;
```

```
    continua = 'S'; // aspas
```

```
    while (continua == 'S')
```

```
        printf("Digite um nu
```

```
        scanf("%d", &numero)
```

```
        resultado = numero *
```

```
        printf("Resultado: %
```

```
        printf("Deseja conti
```

```
        scanf(" %c", &continua); // colocar espaço antes do %
```

```
        printf("----- \n");
```

```
    }
```

```
    return 0;
```

```
}
```

```
Digite um numero inteiro: 2
```

```
Resultado: 6
```

```
Deseja continuar? Digite S para sim ou N para nao: S
```

```
-----
```

```
Digite um numero inteiro: 3
```

```
Digite S para sim ou N para nao: S
```

```
Resultado: 5
```

```
Digite S para sim ou N para nao: N
```

Nesse caso, em vez de um contador, tem-se  
a variável “continua” que armazena a  
**decisão do usuário.**

E se a variável fosse inicializada com um  
conteúdo diferente de ‘S’?

E se o usuário respondesse com ‘s’?

# Estruturas de Repetição – enquanto...faça / while

```
programa {  
    funcao inicio() {  
  
        inteiro numero, resultado  
        caracter continua  
  
        continua = 'S'  
  
        enquanto (continua == 'S') {  
            escreva("Digite um numero inteiro: ")  
            leia(numero)  
            resultado = numero * 3  
            escreva("Resultado: ", resultado, "\n")  
  
            escreva("Deseja continuar? Digite S para sim ou N para nao: ")  
            leia(continua)  
            escreva("----- \n")  
        }  
    }  
}
```

```
Digite um numero inteiro: 2  
Resultado: 6  
Deseja continuar? Digite S para sim ou N para nao: S  
-----  
Digite um numero inteiro: 3  
Resultado: 9  
Deseja continuar? Digite S para sim ou N para nao: N  
-----  
Programa finalizado.
```

# Estruturas de Repetição – enquanto...faça / while

## // EM PSEUDOCÓDIGO

```
programa {  
    funcao inicio() {  
  
        inteiro numero, resultado  
        caracter continua  
        continua = 'S'  
  
        enquanto (continua == 'S') {  
            escreva("Digite um numero inteiro: ")  
            leia(numero)  
  
            resultado = numero * 3  
            escreva("Resultado: ", resultado, "\n")  
  
            escreva("Deseja continuar? Digite S para sim  
ou N para nao: ")  
            leia(continua)  
            escreva("----- \n")  
        }  
    }  
}
```

## // EM LINGUAGEM C

```
#include <stdio.h>  
  
int main() {  
  
    int numero, resultado;  
    char continua;  
    continua = 'S';  
  
    while (continua == 'S') {  
        printf("Digite um numero inteiro: \n");  
        scanf("%d", &numero);  
  
        resultado = numero * 3;  
        printf("Resultado: %d \n", resultado);  
  
        printf("Deseja continuar? Digite S para sim  
ou N para nao: \n");  
        scanf(" %c", &continua);  
        printf("----- \n");  
    }  
    return 0;  
}
```

# Estruturas de Repetição – enquanto...faça / while



Elaborar um programa que efetue a leitura de um valor numérico inteiro qualquer do teclado. Em seguida, multiplique esse valor por 3 e apresente o resultado.

Após apresentar o resultado, o programa deve perguntar ao usuário se ele deseja realizar um novo cálculo. Se a resposta for sim, deve executar novamente.

Ao final da execução do programa, imprima a soma dos resultados obtidos até aquele momento.

# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado, soma;  
    char continua;
```

```
    continua = 'S'; // aspas simples em char  
    soma = 0;
```

```
    while (continua == 'S') {
```

```
        printf("Digite um valor inteiro: ");  
        scanf("%d", &numero);  
        resultado = numero * 3;  
        printf("Resultado: %d \n", resultado);
```

→ (segue no quadro ao lado)

```
        soma = soma + resultado;
```

```
        printf("Deseja continuar? Digite S para  
        sim ou N para nao: ");  
        scanf(" %c", &continua);  
        // colocar espaço antes do % no scanf  
        printf("----- \n");
```

```
    }
```

```
    printf("A soma dos resultados das  
    multiplicacoes eh: %d \n", soma);
```

```
    return 0;
```

```
}
```

# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado, soma;
    char continua;
```

```
    continua = 'S'; // aspas simples
```

```
    soma = 0;
```

```
    while (continua == 'S') {
```

```
        printf("Digite um valor inteiro: \n"),
        scanf("%d", &numero);
        resultado = numero * 3;
        printf("Resultado: %d \n", resultado);
```

```
        soma = soma + resultado;
```

```
        printf("Deseja continuar? Digite S para sim ou N para nao: ");
```

```
        scanf("%c", &continua);
```

*espaço antes do % no scanf*

```
        printf("----- \n");
```

```
        printf("A soma dos resultados das multiplicacoes eh: %d \n", soma);
```

```
Digite um valor inteiro: 2
Resultado: 6
Deseja continuar? Digite S para sim ou N para nao: S
-----
Digite um valor inteiro: 3
Resultado: 9
Deseja continuar? Digite S para sim ou N para nao: S
-----
Digite um valor inteiro: 5
Resultado: 15
Deseja continuar? Digite S para sim ou N para nao: N
-----
A soma dos resultados das multiplicacoes eh: 30
✖
```

# Estruturas de Repetição – enquanto...faça / while

// EM PSEUDOCÓDIGO

```
programa {  
  funcao inicio() {  
  
    inteiro numero, resultado, soma  
    caracter continua  
  
    continua = 'S'  
    soma = 0  
  
    enquanto (continua == 'S') {  
      escreva("Digite um valor inteiro: ")  
      leia(numero)  
      resultado = numero * 3  
      escreva("Resultado: ", resultado, "\n")  
    }  
  }  
}
```

→ (segue no quadro ao lado)

```
soma = soma + resultado  
escreva("Deseja continuar? Digite S para  
sim ou N para nao: ")  
leia(continua)  
escreva("----- \n")  
}  
  
escreva("A soma dos resultados das  
multiplicacoes eh: ", soma, "\n")  
}  
}
```



# Estruturas de Repetição – enquanto...faça / while

// EM PSEUDOCÓDIGO

```
programa {  
  funcao inicio() {
```

```
    inteiro numero, resultado, soma  
    caracter continua
```

```
    continua = 'S'
```

```
    soma = 0
```

```
    enquanto (continua == 'S')  
        escreva("Digite um valor inteiro: ")  
        leia(numero)  
        resultado = numero * 3  
        escreva("Resultado: ", resultado, "\n")
```

→ (segue no quadro ao lado)

```
soma = soma + resultado
```

```
escreva("Deseja continuar? Digite S para  
sim ou N para nao: ")
```

```
leia(continua)
```

```
escreva("----- \n")
```

```
Digite um valor inteiro: 2  
Resultado: 6  
Deseja continuar? Digite S para sim ou N para nao: S  
-----  
Digite um valor inteiro: 3  
Resultado: 9  
Deseja continuar? Digite S para sim ou N para nao: N  
-----  
A soma dos resultados das multiplicacoes eh: 15
```

```
dos resultados das  
h: ", soma, "\n")
```

# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado, soma;  
    char continua;
```

```
    continua = 'S'; // aspas  
    soma = 0;
```

```
    while (continua == 'S')
```

```
    {  
        printf("Digite um valor inteiro: ");  
        scanf("%d", &numero);  
        resultado = numero * 3;  
        printf("Resultado: %d \n", resultado);
```

```
        soma = soma + resultado;
```

```
        printf("Deseja continuar? Digite S para  
        sim ou N para nao: ");  
        scanf(" %c", &continua);  
        // colocar espaço antes do % no scanf  
        printf("----- \n");
```

E se fizéssemos a impressão da variável **soma** dentro da estrutura de repetição?  
Seria impressa a soma parcial entre os resultados das multiplicações.

```
        printf("Os resultados das  
        : %d \n", soma);
```

# Estruturas de Repetição – enquanto...faça / while

// EM LINGUAGEM C

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

```
int main() {
```

```
    int numero, resultado, soma;
```

```
    char continua;
```

```
    continua = 'S'; // aspas simples em char
```

```
    soma = 0;
```

```
    while (continua == 'S') {
```

```
        printf("Digite um valor inteiro: ");
```

```
        scanf("%d", &numero);
```

```
        resultado = numero * 3;
```

```
        printf("Resultado: %d \n", resultado);
```

→ (segue no quadro ao lado)

```
        soma = soma + resultado;
```

```
        printf("Soma parcial: %d \n", soma);
```

```
        printf("Deseja continuar? Digite S para  
sim ou N para nao: ");
```

```
        scanf(" %c", &continua);
```

```
        // colocar espaço antes do % no scanf
```

```
        printf("----- \n");
```

```
    }
```

```
    printf("A soma dos resultados das  
multiplicacoes eh: %d \n", soma);
```

```
    return 0;
```

```
}
```

# Estruturas de Repetição – enquanto...faça / while

```
// EM LINGUAGEM C
#include <stdio.h>
```

```
int main() {
```

```
    int numero, resultado, soma;
    char continua;
```

```
    continua = 'S'; // aspas S
    soma = 0;
```

```
    while (continua == 'S') {
```

```
        printf("Digite um valor inteiro: ");
        scanf("%d", &numero);
        resultado = numero * 3;
        printf("Resultado: %d \n", resultado);
```

```
        soma = soma + resultado;
        printf("Soma parcial: %d \n", soma);
```

```
        printf("Deseja continuar? Digite S para sim ou N para nao: ");
        scanf(" %c", &continua);
```

```
        // Espaço antes do % no scanf
        printf("----- \n");
```

```
    }
    printf("A soma dos resultados das multiplicacoes eh: %d \n", soma);
```

# Para praticar



1. Elaborar um programa que leia dez valores numéricos reais e apresente no final a soma e a média dos valores lidos.
2. Quico tem 1,40 metro e cresce 2 centímetros por ano, enquanto Chaves tem 1,10 metro e cresce 3 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Chaves seja maior que Quico.

# Para praticar



3. Elaborar um programa que leia um número inteiro positivo do teclado e mostre a sua tabuada de 1 a 10.
4. Elabore um algoritmo que efetue a média aritmética de todos os números pares que se encontram no conjunto dos números inteiros de 1 até 100.

# Próxima aula



- Retomada dos principais pontos desta aula
- Estruturas de repetição simples e aninhadas
- Continuidade da introdução à Linguagem C



Universidade Positivo

# Algoritmos de Programação

Aula 11

Prof.<sup>a</sup> Mariane Cassenote

[mariane.cassenote@up.edu.br](mailto:mariane.cassenote@up.edu.br)