

# Estruturas de Repetição

PARTE 2:  
Soma e produto acumulados  
Erros comuns

# Sumário

- Soma acumulada
- Produto acumulado
- Erros comuns em repetições

# Soma de vários números

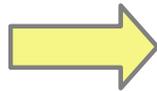
**Problema:** *Uma pessoa deve receber 10 números, somar estes números à medida que forem sendo fornecidos, e após receber os 10 valores, mostrar o valor final desta soma.*

- Normalmente, quem recebe os números para somar, faz o seguinte:
  - ▷ Recebe o 1º número, “anotando” o valor em um pedaço de papel
  - ▷ Recebe o 2º número e soma com o que “anotou no papel” no passo anterior
    - “Anota” a soma **parcial** “no mesmo papel, apagando o valor que tinha antes”
  - ▷ Recebe o 3º número e soma com o que “anotou no papel” no passo anterior
    - “Anota” a soma **parcial** “no mesmo papel, apagando o valor que tinha antes”
  - ▷ Recebe o 4º número e soma com o que “anotou no papel” no passo anterior
    - “Anota” a soma **parcial** “no mesmo papel, apagando o valor que tinha antes”
  - ▷ ...
  - ▷ Recebe o 10º número e soma com o que “anotou no papel” no passo anterior
    - “Anota” a soma **parcial** “no mesmo papel, apagando o valor que tinha antes”
  - ▷ Apresenta a soma

Com excessão do primeiro e último passo, temos um **padrão repetitivo**

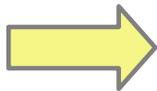
# Soma de vários números

Padrão repetitivo



```
cin >> n;  
soma = soma + n;
```

Controle e  
Teste de parada



```
cin >> n;  
soma = n; // Recebe o primeiro número  
cont = 1; // inicializa contador → já recebeu 1 número  
while ( cont < 10) { // Repetição inicia lendo 2º número  
    cin >> n;  
    soma = soma + n;  
    cont = cont + 1;  
}
```

Incluir a leitura do  
1º número dentro  
da repetição

Soma o 1º número  
com 0 (zero)



```
soma = 0; // inicializa acumulador → Elemento neutro da adição  
cont = 0; // inicializa contador  
while ( cont < 10) { // Repetição inicia lendo 1º número  
    cin >> n;  
    soma = soma + n;  
    cont = cont + 1;  
}
```

# Soma acumulada

Escrever programa que obtém 10 valores do usuário e apresenta a soma destes valores.

```
/* Programa 'Soma10Valores' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int n, soma, cont;  
  
    soma = 0;           // inicializa acumulador → Elemento neutro da adição  
    cont = 0;          // inicializa contador  
    while ( cont < 10) { // Repetição inicia lendo 1º número  
        cin >> n;  
        soma = soma + n;  
        cont = cont + 1;  
    }  
  
    cout << soma << endl;  
    return 0;  
}
```

# Soma acumulada

Escrever programa que soma todos os valores inteiros no intervalo [ **n1**, **n2** ], onde **n1** e **n2** são fornecidos pelo usuário.

```
/* Programa 'escreve_N1aN2' */
```

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
    int n1, n2, k;
```

```
    cin >> n1 >> n2;
```

```
    k = n1;
```

```
    while ( k <= n2 ) {
```

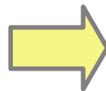
```
        cout << k << endl;
```

```
        k = k + 1;
```

```
    }
```

```
    return 0;
```

```
}
```



```
/* Programa 'SomaIntervalo' */
```

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
    int n1, n2, k, soma;
```

```
    cin >> n1 >> n2;
```

```
    soma = 0; // inicializa acumulador
```

```
    k = n1;
```

```
    while ( k <= n2 ) {
```

```
        soma = soma + k;
```

```
        k = k + 1;
```

```
    }
```

```
    cout << soma << endl;
```

```
    return 0;
```

```
}
```

# Soma acumulada

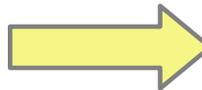
Escrever programa que soma todos os valores inteiros divisíveis por 7 no intervalo [ n1, n2 ], onde n1 e n2 são fornecidos pelo usuário.

```
/* Programa 'SomaIntervalo' */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, k, soma;

    cin >> n1 >> n2;
    soma = 0; // inicializa acumulador
    k = n1;
    while ( k <= n2) {
        soma = soma + k;
        k = k + 1;
    }

    cout << soma << endl;
    return 0;
}
```



```
/* Programa 'SomaDivisiveis7' */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, k, soma;

    cin >> n1 >> n2;
    soma = 0; // inicializa acumulador
    k = n1;
    while ( k <= n2) {
        if ( k % 7 == 0 ) {
            soma = soma + k;
        }
        k = k + 1;
    }

    cout << soma << endl;
    return 0;
}
```

# Soma acumulada

Escrever programa que soma todos os valores inteiros divisíveis por 7 no intervalo [ n1, n2 ], onde n1 e n2 são fornecidos pelo usuário. Além de exibir a soma, deve exibir a **quantidade de valores somados**.

```
/* Programa 'SomaDivisiveis7' */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, k, soma, cont; // precisa de um contador

    cin >> n1 >> n2;
    soma = 0; // inicializa acumulador
    cont = 0; // inicializa contador
    k = n1;
    while ( k <= n2 ) {
        if ( k % 7 == 0 ) {
            soma = soma + k;
            cont = cont + 1; // somente conta se a condição do if é verdadeira
        }
        k = k + 1;
    }

    cout << soma << " " << cont << endl;
    return 0;
}
```

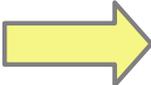
# Produto de vários números

**Problema:** *Uma pessoa deve receber 10 números, multiplicar estes números conforme forem sendo fornecidos, e após receber os 10 valores, mostrar o valor final deste produto.*

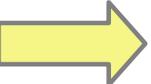
- Normalmente, quem recebe os números para multiplicar, faz o seguinte:
  - ▷ Recebe o 1º número, “anotando” o valor em um pedaço de papel
  - ▷ Recebe o 2º número e multiplica com o que “anotou no papel” no passo anterior
    - “Anota” a produto **parcial** dos 2 números “num pedaço de papel”
  - ▷ Recebe o 3º número e multiplica com o que “anotou no papel” no passo anterior
    - “Anota” a produto **parcial** “no mesmo papel (apaga o valor que tinha antes)
  - ▷ Recebe o 4º número e multiplica com o que “anotou no papel” no passo anterior
    - “Anota” a produto **parcial** “no mesmo papel (apaga o valor que tinha antes)
  - ▷ ...
  - ▷ Recebe o 10º número e multiplica com o que “anotou no papel” no passo anterior
    - “Anota” a produto **parcial** “no mesmo papel (apaga o valor que tinha antes)
  - ▷ Apresenta a produto dos números

Com excessão do primeiro e último passo, temos um **padrão repetitivo**

# Produto de vários números

Padrão repetitivo 

```
cin >> n;  
produto = produto * n;
```

Controle e  
Teste de parada 

```
cin >> n;  
produto = n; // Recebe o primeiro número  
cont = 1; // inicializa contador → já recebeu 1 número  
while ( cont < 10) { // Repetição inicia lendo 2º número  
    cin >> n;  
    produto = produto * n;  
    cont = cont + 1;  
}
```

Incluir a leitura do  
1º número dentro  
da repetição

Multiplica o 1º  
número por 1 (um)

```
produto = 1; // inicializa acumulador → Elemento neutro da  
// multiplicação  
cont = 0; // inicializa contador  
while ( cont < 10) { // Repetição inicia lendo 1º número  
    cin >> n;  
    produto = produto * n;  
    cont = cont + 1;  
}
```

# Produto acumulado

Escrever programa que obtém 10 valores do usuário e apresenta o produto destes valores.

```
/* Programa 'Produto10Valores' */

#include <iostream>
using namespace std;

int main()
{
    int n, produto, cont;

    produto = 1;           // inicializa acumulador → Elemento neutro da multiplicação
    cont = 0;              // inicializa contador
    while ( cont < 10) { // Repetição inicia lendo 1º número
        cin >> n;
        produto = produto * n;
        cont = cont + 1;
    }

    cout << produto << endl;
    return 0;
}
```

# Erros comuns

- Repetição infinita → *loop infinito*
- *Repetição não executada*

# Erros comuns

## Repetição infinita → *loop infinito*

```
/* Programa 'escreve1a10' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int cont;  
  
    cont = 0;           // inicializa contador  
    while ( cont < 10) {  
        cout << cont << endl;  
        cont = cont - 1; // ao invés de somar, subtrai  
                        // Teste de parada sempre será VERDADEIRO  
                        // Repetição não termina nunca  
    }  
  
    return 0;  
}
```

# Erros comuns

## Repetição infinita → *loop infinito*

```
/* Programa 'escreve1a10' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int n, cont;  
  
    cont = 0;           // inicializa contador  
    while ( cont < 10) {  
        cout << cont << endl; // Onde está o incremento / reinicialização?  
                                // Teste de parada sempre será VERDADEIRO  
                                // Repetição não termina nunca  
                                // Exibe sempre o mesmo valor 0 (zero)  
    }  
  
    return 0;  
}
```

# Erros comuns

## Repetição não executa

```
/* Programa 'escreve1a10' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int n, cont;  
  
    cont = 0; // inicializa contador  
    while ( cont > 10) { // Expressão já é falsa de início → Repetição nem começa  
        cout << cont << endl;  
        cont = cont + 1;  
    }  
  
    return 0;  
}
```

## Exercícios para aula *online*

Após assistir todas as vídeo-aulas da semana, procure trabalhar na **Lista de exercícios** do Tópico **Estruturas de Repetição**, na sala virtual da disciplina na UFPR Virtual.

Estes exercícios serão usados nas aulas *online* para esclarecer e consolidar os conceitos abordados até aqui.

# Leitura complementar

Acesse as **Leituras complementares** do Tópico **Estruturas de Repetição**, na sala virtual da disciplina da UFPR Virtual.

Elas são importantes e auxiliam na compreensão dos temas abordados até aqui.

**Créditos:** O conteúdo original deste documento é de autoria da Prof<sup>a</sup> Carmem Satie Hara (DINF/ET), e foi adaptado pelo Prof. Armando L.N. Delgado (DINF/ET) para uso na disciplina *Programação de Computadores* (CI208, CI180, CI183)  
Compartilhe este documento de acordo com a licença abaixo



Este documento está licenciado com uma Licença *Creative Commons* **Atribuição-NãoComercial-SemDerivações** 4.0 Internacional.  
<https://creativecommons.org/licenses/by-nc-sa/4.0/>