

# Estruturas de Repetição

PARTE 3:

Controle por entrada de usuário

# Sumário

- Condição de parada baseada em entrada do usuário

# Ler uma sequência de 10 números

Ler uma sequência de 10 valores do tipo inteiro e para cada valor escrever o seu sucessor.

```
/* Programa 'seqSucessor' */
#include <iostream>
using namespace std;
int main()
{
    int n, k;

    k = 0;
    while ( k < 10 )
    {
        cin >> n;
        cout << n + 1 << endl;
        k = k + 1;
    }

    return 0;
}
```

# Ler uma sequência terminada com zero

Ler uma sequência de valores do tipo inteiro e para cada valor escrever o seu sucessor. Termina quando ler valor 0 (zero), **que não deve ser usado**.

```
/* Programa 'seqSucessor' */
#include <iostream>
using namespace std;
int main()
{
    int n;

    cin >> n;
    cout << n + 1 << endl;
    cin >> n;
    cout << n + 1 << endl;
    cin >> n;
    ...
    cout << n + 1 << endl;
    cin >> n;           // termina quando ler valor 0 (zero)
    return 0;
}
```

# Padrão repetitivo

Qual o padrão repetitivo?

Opção 1

```
cin >> n;  
cout << n + 1 << endl;
```

**OU**

Opção 2

```
cout << n + 1 << endl;  
cin >> n;
```

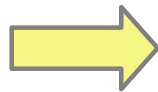
# Opção 1

Padrão repetitivo



```
cin >> n ;  
cout << n + 1 << endl;
```

Controle da  
repetição



```
// inicialização: parece que não precisa  
while ( ... )  
{  
    cin >> n ;  
    cout << n + 1 << endl;  
    // incremento: parece que não precisa  
}
```

# Opção 1

Teste de parada



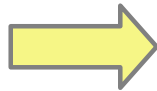
```
// inicialização: parece que não precisa  
while (n != 0) // Termina quando digitar valor 0 (zero)  
{  
    cin >> n ;  
    cout << n + 1 << endl ;  
    // incremento: parece que não precisa  
}
```

Mas pode não funcionar. Por que?

Qual o valor inicial de n?

# Opção 1

Inicialização



```
n = -1; // inicialização ?  
while (n != 0)  
{  
    cin >> n ;  
    cout << n + 1 << endl;  
    // incremento: parece que não precisa  
}
```

Ainda não está bom. Por que?

O que acontece se  
o primeiro valor lido  
é 0 (zero)?



# Padrão repetitivo

Qual o padrão repetitivo?

Opção 1

```
cin >> n;  
cout << n + 1 << endl;
```

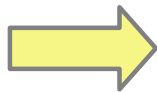
**OU**

Opção 2

```
cout << n + 1 << endl;  
cin >> n;
```

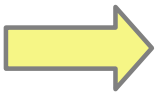
# Opção 2

Padrão repetitivo



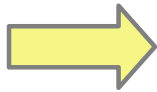
```
cout << n + 1 << endl;  
cin >> n;
```

Controle da  
repetição



```
cin >> n; // inicialização  
while ( ... )  
{  
    cout << n + 1 << endl;  
    cin >> n; // "incremento" → reinicialização  
}
```

Teste de parada



```
cin >> n;  
while (n != 0)  
{  
    cout << n + 1 << endl;  
    cin >> n;  
}
```

# Ler uma sequência terminada com zero

Ler uma sequência de valores do tipo inteiro e para cada valor escrever o seu sucessor. Termina quando ler valor 0 (zero), **que não deve ser usado**.

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

#define FIM 0

int main()
{
    int n;

    cin >> n;
    while ( n != FIM )
    {
        cout << n + 1 << endl;
        cin >> n;
    }
    return 0;
}
```

# Soma de pares

Ler uma sequência de pares de números e, para cada par, escrever a sua soma. A sequência termina quando os valores **0 0** forem digitados.

```
/* Programa 'somaParesValores, v2' */
#include <iostream>
using namespace std;
int main()
{
    int n1, n2;

    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    ...
    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    cin >> n1 >> n2 ; // pára quando for 0 0
    return 0;
}
```

# Programa 'somaParesValores, v2'

Padrão repetitivo →

```
cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;  
cin >> n1 >> n2 ;
```

Controle da  
repetição →

```
cin >> n1 >> n2 ;  
while ( ... )  
{  
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;  
    cin >> n1 >> n2 ;  
}
```

Teste de parada →

```
cin >> n1 >> n2 ;  
while ( (n1 != 0) || (n2 != 0) )  
{  
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;  
    cin >> n1 >> n2 ;  
}
```

# Dois tipos de controle diferente

## Pelo valor lido

```
/* Programa 'somaParesValores' (leitura) */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2;

    cin >> n1 >> n2; // inicialização

    while ( (n1 != 0) || (n2 != 0) )
    {
        cout << n1 + n2 << endl;

        cin >> n1 >> n2; // reinicialização
    }
    return 0;
}
```

## Com contador

```
/* Programa 'somaParesValores' (contador) */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, cont;

    cont = 0; // inicialização

    while ( cont < 20 )
    {
        cin >> n1 >> n2;
        cout << n1 + n2 << endl;

        cont = cont + 1; // reinicialização
    }
    return 0;
}
```

Note a diferença no padrão repetitivo

## 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 Profª 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/>