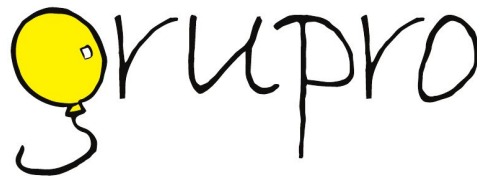




Curso de Programação Nível Básico



Universidade Federal da Bahia
Instituto de Computação
Departamento de Ciência da Computação

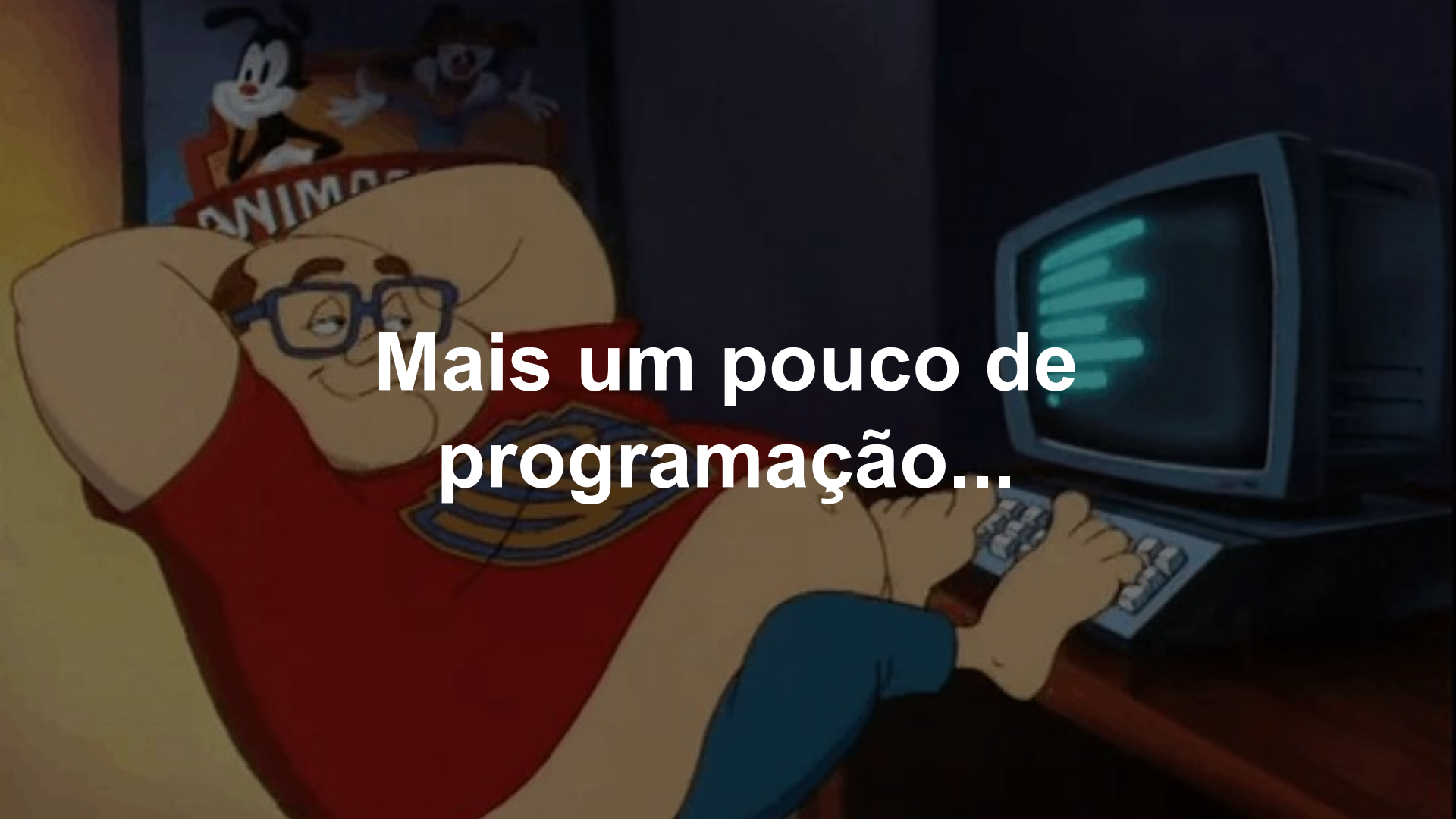
AULA 3 - COMANDOS DE REPETIÇÃO

A cartoon illustration of Donatello, a green turtle with a purple mask and a purple bow on his head. He is sitting at a desk, typing on a laptop keyboard. The background is a dark blue wall with some faint lines. The text "Vamos programar..." is overlaid in the center of the image.

Vamos programar...

Temperatura da semana!

- Descrição
 - Seu programa deve calcular a média aritmética de sete números reais, representando a temperatura em sete dias seguidos e informar se estava quente ou frio. Média acima de 15.0 graus, está quente, e frio, caso contrário.
- Entrada
 - Uma linha contendo sete números reais.
- Saída
 - Seu programa deve imprimir a frase "Quente!" ou "Frio!" seguida de uma quebra de linha.



**Mais um pouco de
programação...**

Como calcular a média da temperatura do mês ou do ano?

- Descrição
 - Seu programa deve calcular a média aritmética da temperatura do ano, ou seja, dos seus 365 dias. Média maior do que 15.0 é considerada quente, e frio, caso contrário.
- Entrada
 - 365 números reais (temperatura) lidos da entrada padrão um para cada dia, sendo um por linha.
- Saída
 - Seu programa deve imprimir a frase "Quente!" ou "Frio!", seguida de uma quebra de linha.

Estrutura de Repetição

Exemplo - comandos de repetição

Mostrar os números pares entre 1 e 50

2 4 6 8 10 12 14 16 18 20 ... 30 ... 40 ... 48

Exemplo - comandos de repetição

Mostrar os números pares entre 1 e 50

2 4 6 8 10 12 14 16 18 20 ... 30 ... 40 ... 48

- **Estado inicial:** 1 ;
- **Ação de repetição:** imprimir número se par;
- **Condição de parada:** ser igual 50 ;
- **Passo:** somar 1 ;

Exemplo - comandos de repetição

Mostrar os números pares entre 1 e 50

2 4 6 8 10 12 14 16 18 20 ... 30 ... 40 ... 48

- **Estado inicial:** 2 ;
- **Ação de repetição:** imprimir número;
- **Condição de parada:** ser igual 50 ;
- **Passo:** somar 2 ;

Estrutura de Repetição For (Para)

```
for (estado inicial ; condição de parada ; passo) {  
    // Ação de repetição  
}
```

Mostrar todos números entre 1 e 100(inclusive)

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:**

```
int i;  
for ( ; ; ) {  
  
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial: $i = 1$;**

```
int i;  
for (i = 1; ; ) {  
  
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:**

```
int i;  
for (i = 1; ; ) {  
  
}
```


Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:** $i \leq 100$;

```
int i;  
for (i = 1; i <= 100; ) {  
  
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:** $i \leq 100$;
- **Passo:**

```
int i;  
for (i = 1; i <= 100; ) {  
  
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:** $i \leq 100$;
- **Passo:** $i++$;

```
int i;  
for (i = 1; i <= 100; i++) {  
  
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:** $i \leq 100$;
- **Passo:** $i++$;
- **Ação de repetição:**

```
int i;
```

```
for (i = 1; i <= 100; i++) {
```

```
}
```

Mostrar todos números entre 1 e 100(inclusive)

- **Estado inicial:** $i = 1$;
- **Condição de parada:** $i \leq 100$;
- **Passo:** $i++$;
- **Ação de repetição:** mostrar o número.

```
int i;  
for (i = 1; i <= 100; i++) {  
    cout << i << endl;  
}
```

Mostrar todos números entre 1 e 100 (inclusive)

```
int i;  
for (i = 1; i <= 100; i++) {  
    cout << i << endl;  
}
```

OU

```
int i;  
for (i = 1; i <= 100; i++)  
    cout << i << endl;
```

Comandos de repetição - Desafios



Divisíveis por 2 e 3

- `divisiveis.cpp`
- Descrição
 - Seu programa deve imprimir os números divisíveis por 2 e também por 3 que sejam menores que N.
- Entrada
 - Um número N indicando o limite para impressão.
- Saída
 - Seu programa deve imprimir os números que são divisíveis por 2 e 3 ao mesmo tempo. Imprima um por linha.

Mostrar os números que são divisíveis por 2 e 3 menores que 100.

```
int i;  
for ( i = 1 ; i<100; i++){  
    if ( i % 2 == 0 && i % 3 == 0 ) {  
        cout << i << endl;  
    }  
}
```

**Mostrar os números que são divisíveis por
2 e 3 menores que 100.**

```
int i;  
for ( i = 6 ; i<100 ; i+=6 ){  
    cout << i << endl;  
}
```

Indicação da quantidade de números da entrada - Problemas usando FOR

Programa dobro

Dada uma lista de **N** inteiros, imprima o dobro de cada número.

Entrada

A entrada é composta de várias linhas, cada um com um número. O primeiro número é um inteiro positivo **N**. Os **N** números seguintes são valores inteiros **X** para os quais se deve calcular o dobro.

Saída

Seu programa deve produzir **N** linhas na saída, contendo o dobro de cada um dos **N** números.

```
int n, x;

cin >> n;

for (int i=0; i<n; i++){

    cin >> x;

    cout << x * 2 << endl;

}
```

Entrada	Saída
2 2 3	4 6
3 1 5 3	2 10 6

Loops aninhados

As estruturas de repetição são consideradas instruções

Isso significa que é possível incluir uma estrutura de repetição dentro de outra

Isso é o que chamamos de **loops aninhados**

for - Repetição de contagem

Exemplo: programa que imprime a tabuada de 9

```
for (int j=1; j<=10; i++)
```

```
    cout << "9 * " << j << " = " << 9 * j << endl;
```

Pensando “de fora pra dentro”

```
# escrever a tabuada de 1 até a tabuada de 10
```

```
for (int i=1; i<=10; i++)
```

```
    // escrever a tabuada de i
```

```
        for (int j=1; j<=10; j++)
```

```
            cout << i << “ * ” << j << “ = ” << i * j << endl;
```

Comandos de repetição



E se a quantidade de entradas
não for conhecida?

while - Repetição pré-testada

inicialização variável iterativa

while (condição)

comando

atualização da variável iterativa

comando único

OU

inicialização variável iterativa

while (condição){

comando1

...

comandoN

atualização da variável iterativa

sequência de comandos

}

Valor especial (sentinela) indicando fim da entrada - Problemas com WHILE

Dada uma sequência de números inteiros, imprima o dobro de cada número.

Entrada

A entrada é composta de várias linhas, cada uma com um número da sequência. O número 0 representa o final da entrada; isto é, ele não faz parte da sequência e, ao ler esse valor, seu programa deve encerrar.

Saída

Para cada linha da entrada (exceto o 0), seu programa deve imprimir uma linha com o dobro do número lido.

```
cin >> x;
while (x != 0){
    cout << x * 2 << endl;
    cin >> x;
}
```

Entrada	Saída
1 2 3 0	2 4 6
2 -1 0	4 -2

while - Repetição pré-testada

Exemplo: imprimir de 1 a 3 e imprimir “Fim”

```
i = 1 // 1. inicialização da variável de controle  
while (i <= 3){ // 2. teste da condição  
    cout << i << endl; // 3. instruções a serem repetidas  
    i = i + 1 // 4. atualização da variável de controle  
}
```



While

- Qual a saída do código abaixo ?:

```
t = 5;
while (t > 0){
    cin >> r;
    r -= 1;
    cout << r << endl;
}
cout << "Final" << endl;
```

Encontre o erro!

JUST WAITING



**FOR YOUR INFINITE LOOP TO
END**

memegenerator.net



While

```
while (i <= 3){
```

```
    i = 1;
```

```
    cout << i << endl;
```

```
    i = i + 1;
```

```
}
```

```
print("Fim")
```

Encontre o erro!

While (programa para contar de 3 a 8)

```
i = 3;
```

```
while (i > 8){
```

```
    cout << i << endl;
```

```
    i = i + 1;
```

```
}
```

```
cout << "Fim" << endl;
```

Encontre o erro!

break e continue

As instruções **break** e **continue** permitem alterar o fluxo de execução normal de um loop (seja **for** ou **while**)

Observações:

- Em geral é possível escrever programas sem usar essas instruções
- Em geral o seu uso deve ser evitado, pois tende a tornar o código mais difícil de entender (sobretudo em loops com muitas linhas de código)

break

Dentro de qualquer estrutura de repetição, é possível usar a instrução **break** para **sair do loop**

```
while (True){  
    cin >> n;  
    if (n == 0)  
        break;  
    cout << n * 2 << endl;  
}  
cout << "Fim" << endl;
```

continue

Dentro de qualquer estrutura de repetição, é possível usar a instrução **continue** para **voltar para o início do loop**

```
# Não imprime os números pares
for (int i=1; i<=100; i++){
    if (i % 2 == 0)
        continue;
    cout << i << endl;
}
```

Problema SOMA - adaptado de <https://br.spoj.com/problems/SOMA/>

Dada uma lista de N inteiros, encontre a soma de todos eles.

Entrada

A entrada é composta de várias linhas, cada uma com um número. O primeiro número é um inteiro positivo N . Os N números seguintes são valores inteiros x que devem ser somados.

Saída

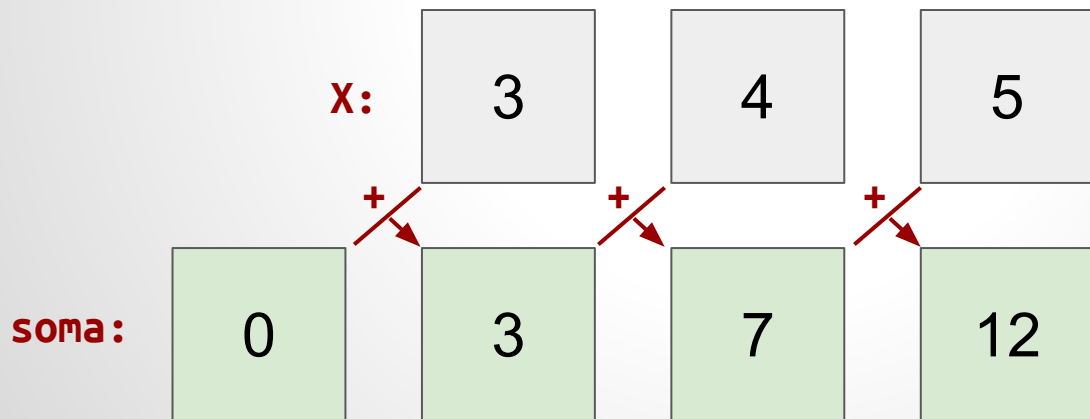
Seu programa deve produzir uma única linha na saída, contendo a soma de todos os N inteiros.

Entrada	Saída
2 2 3	5
3 1 5 3	9

```
cin >> n;  
  
soma = 0;  
  
for (int i=0; i<n; i++){  
    cin >> x;  
    soma += x;  
}  
  
cout << soma << endl;
```

Conceito: acumulador

Acumulador é uma variável que guarda o resultado parcial de uma computação, sendo atualizada dentro de um loop com base no seu valor anterior e no valor do item sendo analisado na iteração atual. É o caso da variável **soma** do exemplo anterior:



Problema MÁXIMO - adaptado de <https://br.spoj.com/problems/SOMA/>

Dada uma lista de N inteiros não-negativos distintos entre si, encontre o maior de todos eles.

Entrada

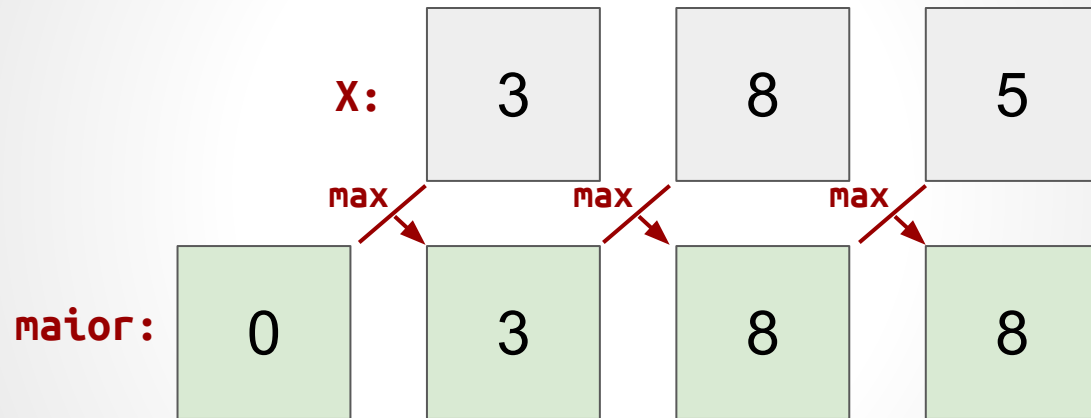
A entrada é composta de vários números em linhas diferentes. O primeiro número é um inteiro positivo N . Os N números seguintes são valores inteiros X a serem analisados.

Saída

Seu programa deve produzir uma única linha na saída, contendo o maior de todos os N inteiros.

Entrada	Saída
2 2 3	3
3 1 5 3	5

```
cin >> n;
maior = -1;
for (int i=0; i<n; i++){
    cin >> x;
    if (x > maior)
        maior = x;
}
cout << maior << endl;
```



Outros desafios

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1080>

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1143>

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1151>

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1153>

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1165>

Onde estudar?

GruPro - UFBA

<http://maratona.dcc.ufba.br>

Onde praticar?

Spoj

<http://br.spoj.com/>

URI

<https://www.urionlinejudge.com.br/judge/pt/>

OBI

<http://olimpiada.ic.unicamp.br/pratique>