



# Escola de Engenharia - FCI

## Algoritmos e Programação I



### Laboratório 07



Looping “while”, “for”, validação, repetição,  
sequência e série

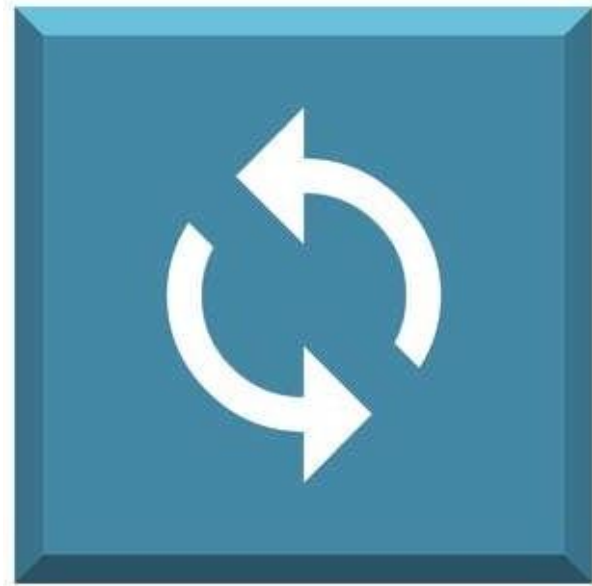
Pror. Ubirajara Carnevale de Moraes  
Profa. Melanie Lerner Grinkraut

# Material para consulta



# Comandos de Looping

- **while ()**
- for ()

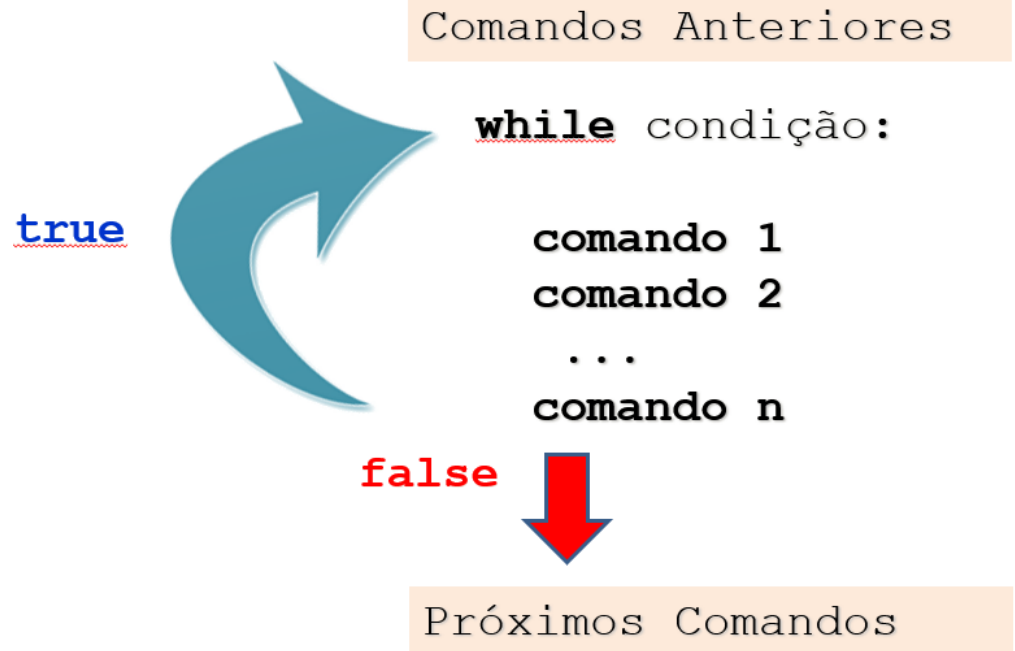


# Com Repetição (Loop)

Elaborar um programa para exibir os **números inteiros entre 1 e 15**. Depois, exibir a palavra “FIM”

```
n=1
while n<=15:
    print(n)
    n=n+1
print("FIM")
```

## Sintaxe do comando WHILE



# Material para consulta



# Aplicação do Looping while

Consistência de Dados (Validação)



<https://img.freepik.com/>



# Codificação quase completa....

```
import math as m
```

```
R=0
```

```
while R<=0:
```

```
    R= float (input ("Raio: "))
```

```
h=0
```

```
while h<=0:
```

```
    h= float (input ("Altura: "))
```

```
S= 2*m.pi*R*(R+h)
```

```
V= m.pi*m.pow(R,2)*h
```

```
print(" Área Total= ", S)
```

```
print(" Volume= ", V)
```



**Lembre-se:**  
A condição será  
sempre **o erro !!**

# Mensagens de Erro na digitação incorreta !!





# Área e Volume do Cilindro Reto

```
import math as m
```

```
R=0
```

```
while R<=0:
```

```
    R= float (input ("Raio: "))
```

```
    if R<=0:
```

```
        print(" Raio inválido!")
```

```
h=0
```

```
while h<=0:
```

```
    h= float (input ("Altura: "))
```

```
    if h<=0:
```

```
        print(" Altura inválida!")
```

```
S= 2*m.pi*R*(R+h)
```

```
V= m.pi*m.pow(R,2)*h
```

```
print(" Área Total= ", S)
```

```
print(" Volume= ", V)
```

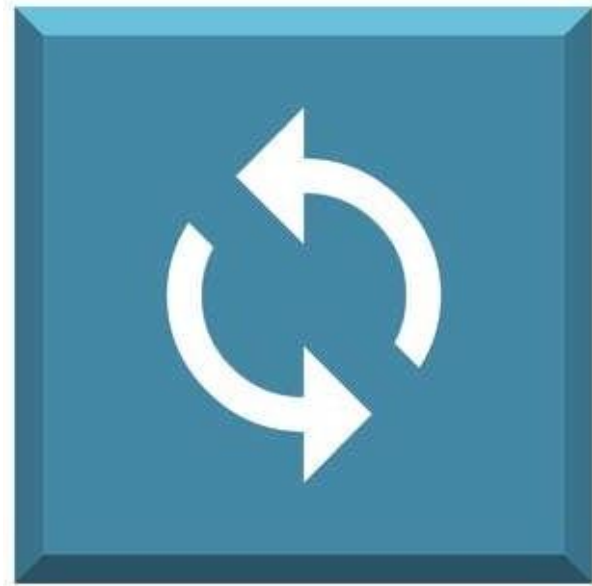


# Material para consulta



# Comandos de Looping

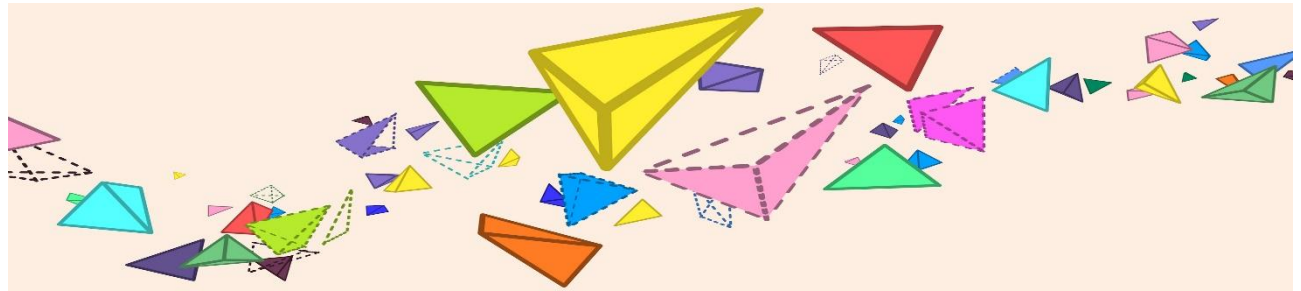
- **while ()**
- **for ()**



# Aplicações do comando for ()

1) **Gerar intervalos de números:** 1, 2, 3, 4 ou  
20, 22, 24, 26 ou 0, -1, -2

2) **Repetição de tarefas:** calcular a área de 20 triângulos



3) Exibir uma **sequência com n elementos:** 3, 6, 9,...

4) Calcular **uma somatória:**  $3 + 6 + 9 + 12 + \dots$

# Sintaxe do comando for

```
for variável in range(valor inicial, limite final) :
```

```
comando 1
```

```
comando 2
```

```
...
```

```
comando n
```

```
comando n+1
```

Inicia nesse  
valor

Vai até o  
valor  
anterior

Após executar o  
for, continua  
para as linhas  
seguintes a ele



# Sintaxe do comando for ()

Inicia nesse  
valor

Vai até o valor  
anterior

Razão. Quando  
ausente, o default  
é 1 (padrão)

```
for variável  
in range(valor inicial, limite final, incremento) :  
    comando 1  
    comando 2  
    ...  
    comando n
```

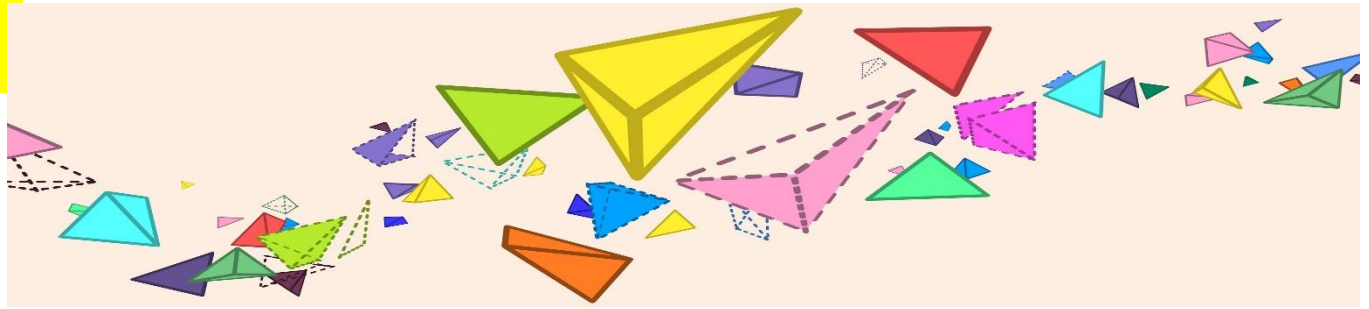
Após executar o  
for, continua  
para as linhas  
seguintes a ele



# Aplicações do comando for ()

1) Gerar intervalos de números: 1, 2, 3, 4 ou 20, 22, 24, 26 ou 0, -1, -2

2) Repetição de tarefas: calcular a área de 20 triângulos



3) Exibir uma sequência com n elementos: 3, 6, 9,...

4) Calcular uma somatória:  $3 + 6 + 9 + 12 + \dots$

```
1 # Área de N Triângulos
2 # Validação de N
3 N=-1
4 ▼ while N<=0:
5     N = int (input("Digite o número de triângulos:"))
6 ▼     if N<=0:
7         print ("Valor de N inválido\n")
8
9 ▼ for i in range (1, N+1):
10     b=-1
11 ▼     while b<=0:
12         b= float (input ("Base: "))
13 ▼         if b<=0:
14             print ("Valor da base incorreto\n")
15     h=-1
16 ▼     while h<=0:
17         h= float (input ("Altura: "))
18 ▼         if h<=0:
19             print ("Valor da altura incorreto")
20     A= b*h/2
21     print(" Área do Triângulo= ", A, "\n")
22 print ("Fim")
```

Diagram illustrating the execution flow of the Python code for calculating the area of N triangles, with steps numbered 1 through 8:

- Initialization: `N = -1`
- Validation Loop: `while N <= 0:` (Step 2)
- Invalid Input Handling: `if N <= 0:` and `print ("Valor de N inválido\n")` (Step 3)
- Triangle Loop: `for i in range (1, N+1):` (Step 4)
- Base Validation Loop: `while b <= 0:` (Step 5)
- Base Input Handling: `b = float (input ("Base: "))` and `if b <= 0:` (Step 6)
- Base Error Handling: `print ("Valor da base incorreto\n")` (Step 7)
- Height Validation Loop: `while h <= 0:` (Step 8)



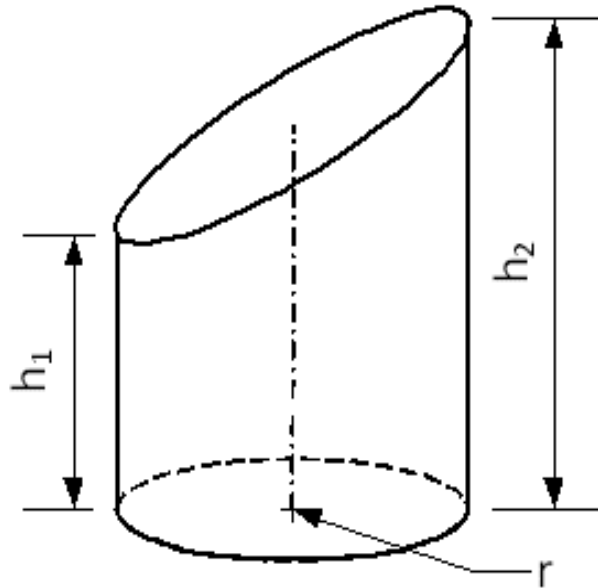
# Observe na codificação anterior ...

- 1 Iniciar a variável com um valor errado e o while
- 2 Fazer a entrada para o valor de N
- 3 Exibir a mensagem de erro com o if
- 4 Abrir o “for” para a repetição de N vezes
- 5 Fazer a validação com while (condição de erro)
- 6 Mais de um campo de entrada, repetir o while
- 7 Depois da validação dos campos, calcular.
- 8 Após o “for” continuar o programa ou finalizar.

**1) Exercício 12.30** Elaborar um programa para calcular a Área Lateral (A) e o Volume (V) de **n** cilindros com secção oblíqua. **Fazer consistência (validação dos dados de entrada e exibir mensagem de erro personalizada).**

$$A = \pi \cdot r \cdot (h_1 + h_2)$$

$$V = \pi \cdot r^2 \cdot \frac{h_1 + h_2}{2}$$



<https://chrissainty.com/content/images/2019/02/global-error-handling.jpg>

2) Elaborar um programa para, a partir de 3 notas, calcular a média aritmética e a média ponderada de **n** alunos.

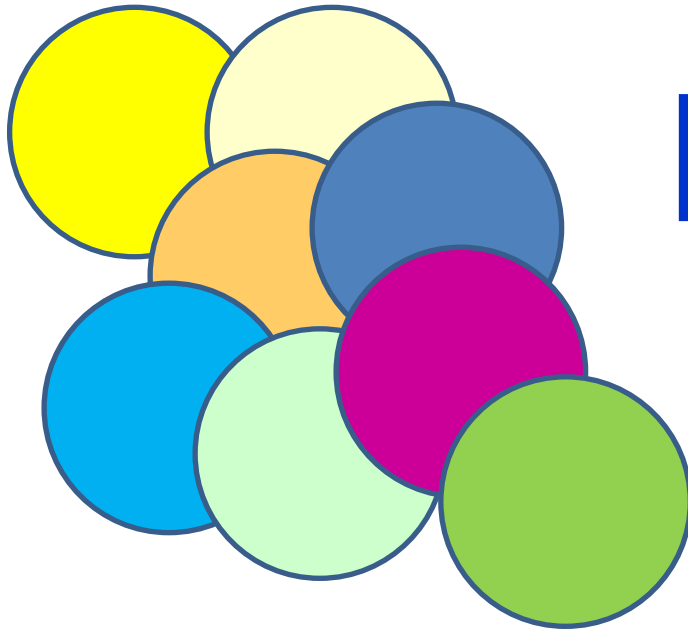
**Fazer consistência (validação) nos campos de entrada com mensagens de erro personalizadas.**

Sabe-se que:

$$\text{Média Ponderada} = \frac{N1*2 + N2*2 + N3*6}{10}$$



3) 7.22 Elaborar um programa para calcular a área e o comprimento da circunferência de “N” círculos. Lembre-se: Consistência e mensagem de erro nas entradas de dados!!



$$a = \pi r^2 \quad c = 2\pi r$$



OBS.: Entrada de Dados com validação e mensagem de erro

## 4) Alterar o programa codificado no LAB 6 com **Menu de Opções**, fazendo validação (Consistência) nos dados de entrada (opção) com mensagem de erro.

\* QUATRO OPERAÇÕES ENTRE DOIS VALORES \*

MENU:

(1) Adição

(2) Subtração

(3) Multiplicação

(4) Divisão

Opção:

xxxxxxxxxxxxxxxxxx

Primeiro Valor:

Segundo Valor:

xxxxxxxxxxxxxxxxxx

Resultado=



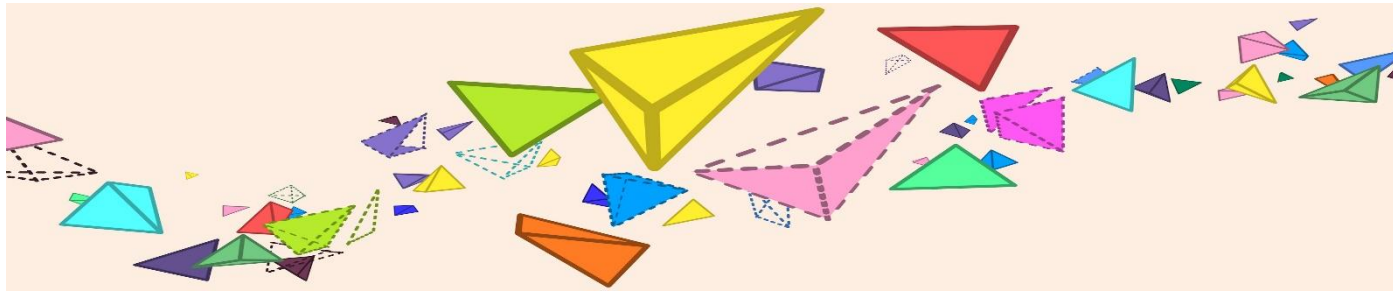
**Atenção:** uma das opções de operação aritmética também tem restrição matemática. Qual é?

Faça a validação com mensagem de erro.

# Aplicações do comando for ()

1) **Gerar intervalos de números:** 1, 2, 3, 4 ou  
20, 22, 24, 26 ou 0, -1, -2

2) **Repetição de tarefas:** calcular a área de 20  
triângulos



3) **Exibir uma sequência com n elementos:** 3, 6, 9,...

4) **Calcular uma somatória:**  $3 + 6 + 9 + 12 + \dots$

# Criar uma sequencia com $N$ elementos:



- Obs.: não sabemos o último valor. Mas sabemos o valor de  $N$  e também podemos identificar a razão da sequência (cinco).

## Ex1: Exibir na tela a sequência 3 5 7 9 11... para N elementos

Neste caso, precisaremos verificar o **termo geral** da sequência.

Observar atentamente a sequência.

Veja qual é a **razão entre os elementos**. A partir da razão conseguiremos definir o termo geral que neste caso é:

$$2 * K + 1$$

Mas de onde tiramos esses valores?



A sequência **3 5 7 9 11...** para **n elementos**

*Termo Geral:  **$2*K+1$***

O termo geral em azul foi obtido da seguinte forma:

***Variável usada no comando “for” (K) multiplicada pela razão da sequência (2) +/- ajustes (1)***

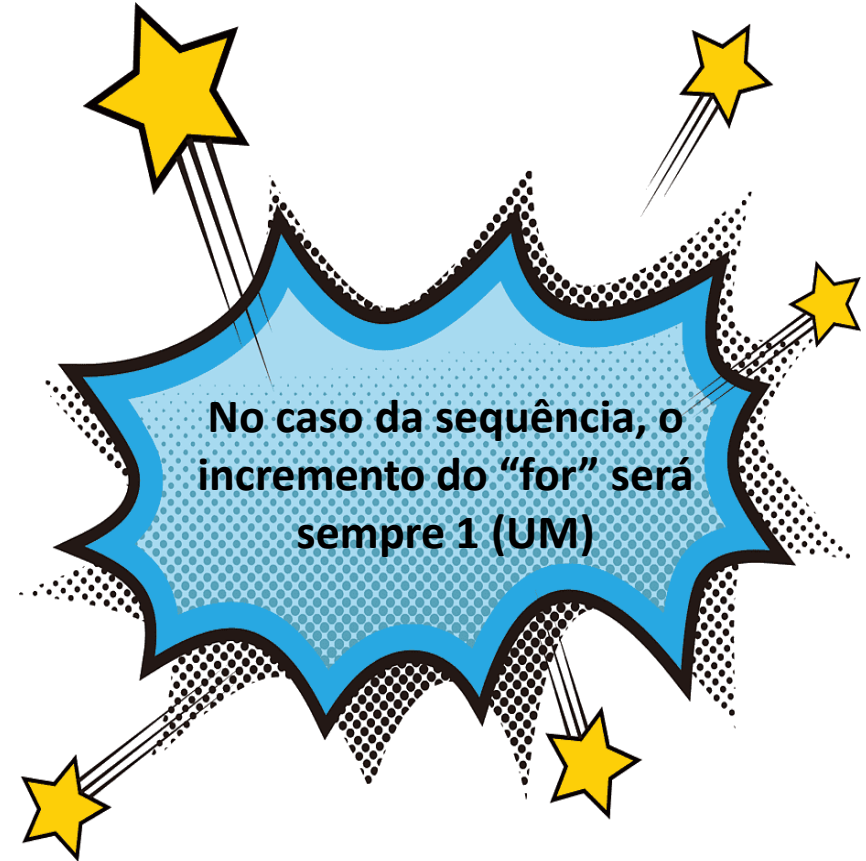
- Realmente a razão é 2 que multiplicamos pela variável que será usada no “for”, ou seja  $2*K$ , porém  $2*k$ , com o K iniciando de 1 (1º valor) resulta em  $2*1=2$ . Porém o primeiro número é 3. Então é necessário somar 1 nos ajustes para se chegar ao 3.
- É possível testar, substituindo o K pelos valores que ele ira assumir (1, 2, 3, ...n).... Os valores exibidos na tela serão  $2*1+1$  ,  $2*2+1$  ,  $2*3+1$ , etc, ou seja, 3, 5, 7 e assim por diante.

Exibir na tela a sequência **3 5 7 9 11...** para **n** elementos.

Usaremos então o Looping “for”. Assim:

```
for K in range (1, N+1):  
    print (2*K+1)
```

  
**Termo Geral**





# Codificação completa:

```
N=-1
while N<=0:
    N = int(input("Digite o valor de N:"))
    if N<=0:
        print("Valor de N inválido")
for K in range(1, N+1):
    print(2*K+1)
```

A sequência **3 5 7 9 11...** para **n elementos** será exibida na tela:

```
Digite o valor de N:5
3
5
7
9
11
> □
```

2) Exibir na tela a sequência **2, 5, 8, 11, 14...**  
para **n elementos**

3) Exibir na tela a sequência **1, 4, 7, 10,...**  
para **n elementos**

4) Exibir na tela a sequência  **$5x, 6x^2, 7x^3, \dots$**   
para **n elementos**

# Atividade Complementar



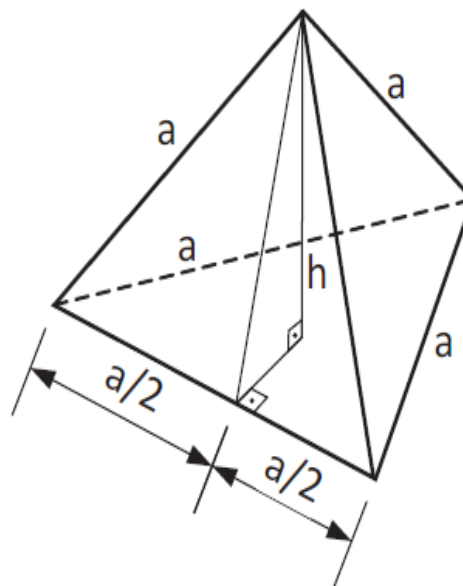
7.30	<b>Dados:</b> $n = 3$ ,	
	$a = 5$	<b>Resposta:</b> $h = 4.08248$ , $A = 43.3013$ e $V = 14.7314$
	$a = 2$	<b>Resposta:</b> $h = 1.63299$ , $A = 6.9282$ e $V = 0.942809$
	$a = 3.8$	<b>Resposta:</b> $h = 3.10269$ , $A = 25.0108$ e $V = 6.46673$

**7.30** Dado o valor da Aresta  $a$  de  $n$  tetraedros regulares, elaborar um programa C++ para calcular e exibir a Altura  $h$ , a Área  $A$  e o Volume  $V$  de cada tetraedro.

$$h = \frac{a\sqrt{6}}{3}$$

$$A = a^2\sqrt{3}$$

$$V = \frac{a^3\sqrt{2}}{12}$$



Algoritmos e Linguagem

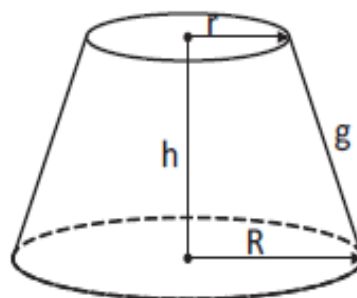
C++

Sergio Vicente D. Pamboukian  
Lincoln Cesar Jamboni  
Edson A. R. Barros

7.35	Dados: $n = 2$ ,	
	$R = 5, r = 2, h = 2$	Resposta: $V = 81.6814$
	$R = 10, r = 3, h = 4$	Resposta: $V = 582.242$

**7.35** Criar uma função em Linguagem C++ para calcular o Volume (V) de um Tronco de Cone Reto de bases paralelas. A função deve receber como parâmetros o raio da base maior (R), o raio da base menor (r) e a altura (h) do tronco e devolver o valor do Volume. Não utilize cin/cout nesta função.

$$V = \frac{1}{3} \cdot \pi \cdot h \cdot (R^2 + R \cdot r + r^2)$$



Em seguida, elaborar um programa (função *main*) para calcular o volume de  $n$  Troncos de Cone. O valor de  $n$  e os  $n$  valores de  $R$ ,  $r$  e  $h$  devem ser fornecidos pelo usuário. Cada volume deve ser calculado utilizando obrigatoriamente a função criada anteriormente. Os volumes calculados devem ser exibidos na tela.

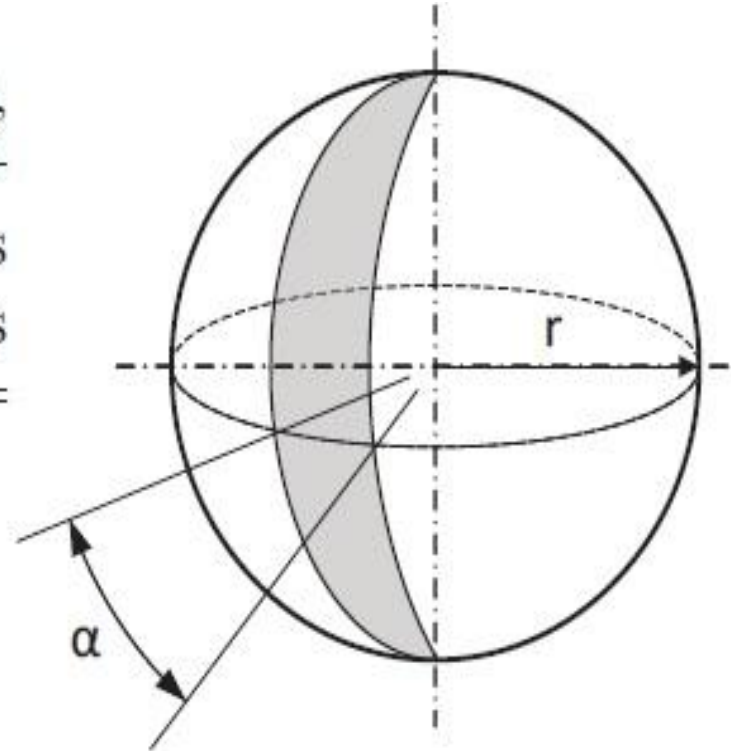
Algoritmos e Linguagem

C++

Sergio Vicente D. Pamboukian  
Lincoln Cesar Zamboni  
Edson A. R. Barros

**7.65** Dado o raio ( $r$ ) de uma esfera, elaborar um programa C++ para calcular e exibir as áreas ( $A$ ) dos fusos esféricos correspondentes quando  $\alpha = 10^\circ, 20^\circ, 30^\circ, \dots 360^\circ$ .

$$A = \frac{2\pi r^2 \alpha}{180}$$





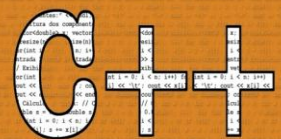
8.13	Dados: $n = 5$ e $x = 2$	Resposta: 2, 8, 32, 128 e 512
------	--------------------------	-------------------------------

**8.13** Dados os valores  $x$  e  $n$ , elaborar uma função de usuário para exibir as  $n$  primeiras parcelas da sequência:

$$x, \quad x^3, \quad x^5, \quad \dots$$

Escrever também a função principal (*main*) que contém a chamada da função de usuário.

Algoritmos e Linguagem



Sergio Vicente D. Pamboukian  
 Lincoln Cesar Zamboni  
 Edson A. R. Barros

## 9 Séries Polinomiais

Combinando as duas somatórias...

Handwritten mathematical derivation showing the combination of two series into a single polynomial series.

Top series:  $S = 2 + 5 + 8 + 11 \dots$

General term for top series:  $\boxed{\text{Termo Geral } 3i-1}$

Bottom series:  $S = X + X^2 + X^3 + X^4 \dots$

General term for bottom series:  $\boxed{\text{Termo Geral } X^i}$

Combined series:  $S = 2 \times X + 5 \times X^2 + 8 \times X^3 + 11 \times X^4 \dots$

General term for combined series:  $\text{Termo Geral} \Rightarrow (3i-1) \times X^i$

$$S = \frac{2.X}{3} + \frac{5.X^2}{4} + \frac{8.X^3}{5} + \frac{11.X^4}{6} + \dots$$

$$\text{Ex 1)} S = 2.X + 5.X^2 + 8.X^3 + 11.X^4 + \dots$$

$$\text{Ex 2)} S = 2.X - 5.X^2 + 8.X^3 - 11.X^4 + \dots$$

$$\text{Ex 3)} S = -2.X + 5.X^2 - 8.X^3 + 11.X^4 + \dots$$

$$\text{Ex 4)} \quad S = -\frac{2x}{3} + \frac{3x^2}{5} - \frac{4x^3}{7} + \dots$$

$$\text{Ex 5)} S = 1! + 2! + 3! + 4! + 5! \dots$$

$$\text{Ex 6)} \quad \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$



# Universidade Presbiteriana Mackenzie

## Escola de Engenharia - FCI



**Bom fim de semana e bons estudos!!**