BIS0005 - Bases Computacionais da Ciência

Aula 05 - Lógica de programação: variáveis e estruturas sequenciais.

Saul Leite
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

Q2 2018

Introdução

Programa é uma sequência de ordens (comandos, instruções) dadas a um computador que, a partir de dados inseridos, obtêm um resultado que será disponibilizado por algum dispositivo de saída.

Instrução é a informação que indica a um computador uma ação elementar.

Para a construção de um programa é necessário um conjunto de instruções colocadas em ordem sequencial lógica: **Algoritmo**.

Algoritmo

Algoritmo: passos executados em sequência lógica até atingir um objetivo ou solução de um problema.

Exemplos

- Fazer um bolo
- Construir um robô para explorar um local desconhecido
- Trocar uma lâmpada

Algoritmo

Algoritmo: passos executados em sequência lógica até atingir um objetivo ou solução de um problema.

Exemplos

- Fazer um bolo
- Construir um robô para explorar um local desconhecido
- Trocar uma lâmpada

Obs.: O programa é a realização (ou implementação) do algoritmo em um computador específico.

Origem do nome Algoritmo

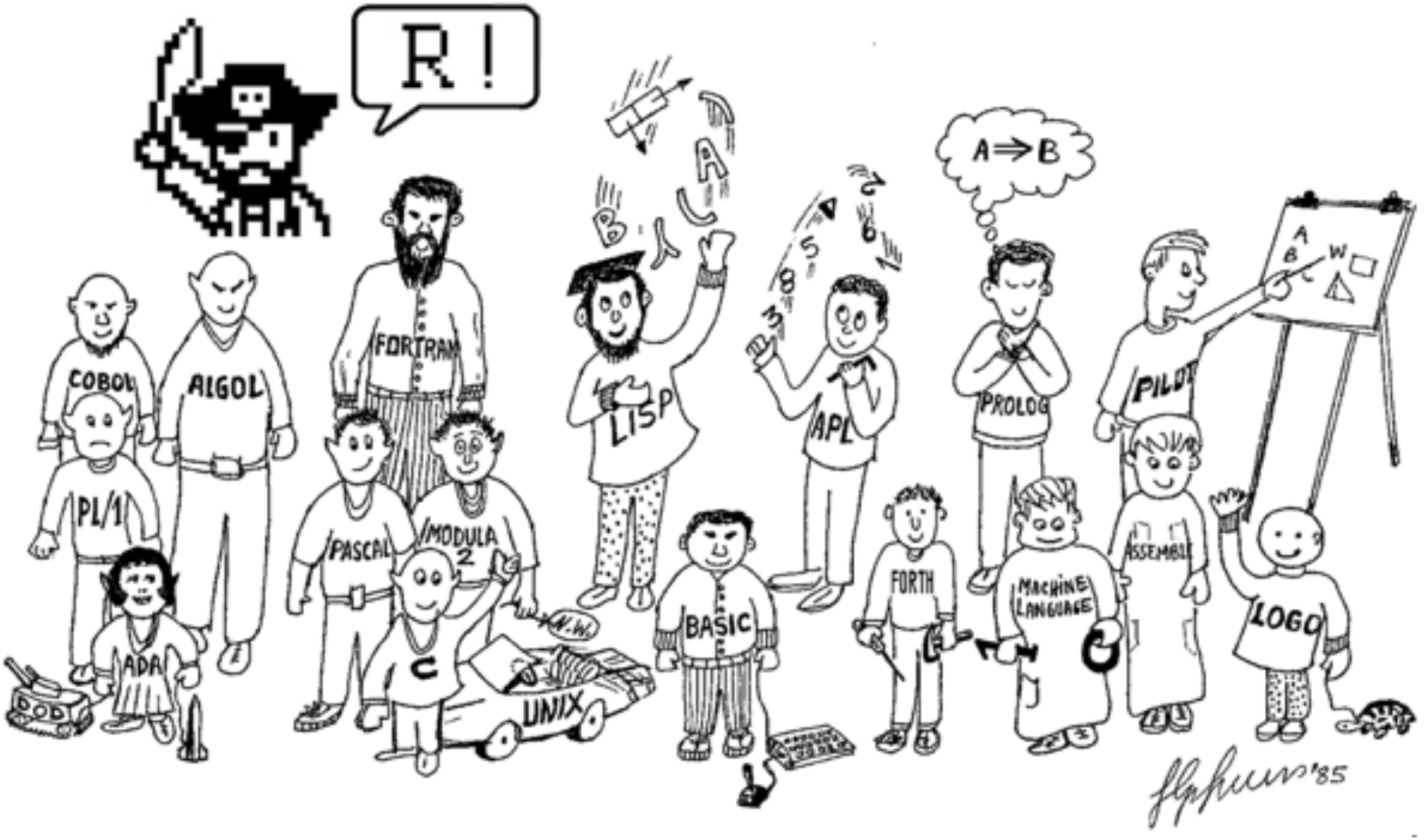
Algoritmos na matemática têm uma longa história:

1. **Babilônia** para calcular inversos de números inteiros (encontrado em bloco de barro VAT 6505, datado aprox. 2000 A.C. - 1650 AC)
2. **Egípcios** para fazer aritmética (Rhind Papyrus - 1650 AC)
3. **Gregos**, como no algoritmo de Euclides (para encontrar MDC - 300 AC) e Crivo de Eratóstenes (para listar os números primos - 200 AC).

O nome Algoritmo é uma versão Latina do nome Al-Khwarizmi, um matemático Persa (780 - 850), seu livro "al-Jabr wa I-Muqabala" foi origem do nome Álgebra. A palavra Algarismo, também derivado do seu nome.

Linguagens de programação

Linguagens de programação



Tipos básicos de variáveis (em R)

integers: -3,-2,1,0,1,2,3 (guada números inteiros).

numeric: números fracionários (idêntico a **double**).

character: para textos.

logical: variáveis que representam TRUE ou FALSE.

complex: para números complexos (que possuem parte imaginária).

Outras linguagens possuem tipos análogos a estes, mas a nomenclatura pode variar.

Atribuindo valores a variáveis

Declaração de variáveis:

Criando uma variável com valor inteiro: (O L no final do número indica que você deseja a representação do número como inteiro, caso contrário o R assumirá como *numeric*.)

```
x <- 3L
```

Criando uma variável com valor *numeric*:

```
y <- 3.1415
```

Criando uma variável de tipo *charater*: (**Atenção** o texto deve vir entre aspas.)

```
z <- "Bases Computacionais"
```

Criando uma variável com valor lógico:

```
w <- TRUE
```

Atribuindo valores a variáveis

Para Verificar o tipo da variável, podemos usar o comando **class**.

```
class(x)
```

```
## [1] "integer"
```

```
class(y)
```

```
## [1] "numeric"
```

```
class(z)
```

```
## [1] "character"
```

```
class(w)
```

```
## [1] "logical"
```

Ambiente Global

Em uma sessão padrão, o R mantém todas as variáveis declaradas no **ambiente global** ("Global Environment"). - *exceto quando declaradas dentro de funções, como veremos mais a frente*.

Variáveis no ambiente global podem ser exibidas com o comando **ls()**, e removidas com o comando **rm()**. Exemplo, para apagar todas as variáveis, basta chamar

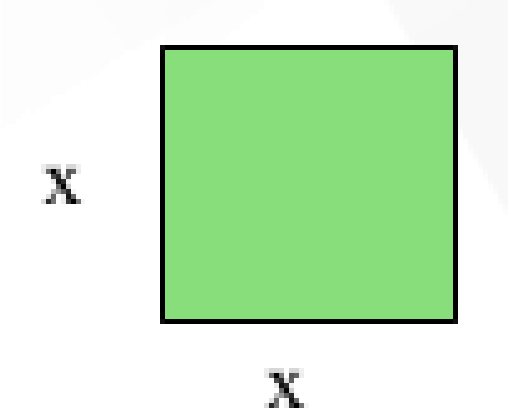
```
# Remove todas as variáveis do ambiente global.
rm(list=ls())
```

Operações aritméticas

Operações	Descrições
+ x	Operação unária (positivo)
- x	Operação unária (negativo)
x + y	Soma
x - y	Subtração
x * y	Multiplicação
x / y	Divisão
x ^ y	Potência
x %% y	Resto da divisão inteira
x %/% y	Divisão inteira (não guarda parte fracionária)

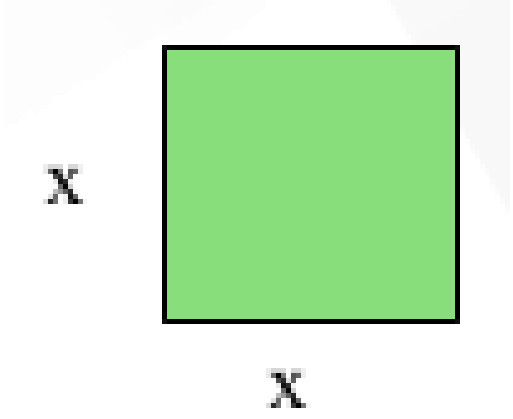
Operações aritméticas: Exemplo

Cálculo da área, perímetro e diagonal de um quadrado de lado x.



Operações aritméticas: Exemplo

Cálculo da área, perímetro e diagonal de um quadrado de lado x.



Desta vez, vamos montar um **programa**, portanto, usando o RStudio, escolha no menu File $\rightarrow$ New File $\rightarrow$ R Script.

Obs.: O termo *script* é frequentemente usado para se referir a programas escritos em linguagens interpretadas como o R, Python, Matlab, . . . , já que estas linguagens não geram código de máquina.

Operações aritméticas: Exemplo

Cálculo da área, perímetro e diagonal de um quadrado de lado x. O seguinte código deve ser inserido no seu *script*:

```
#Atribuindo valor para o lado do quadrado
x <- 2

#Calculando a área
area <- x^2

#Calculando o perímetro
perimetro <- 4*x

#Calculando a diagonal
diagonal <- x*sqrt(2)
```

Obs.: Todo texto inserido após o símbolo # é ignorado pelo interpretador do R. Isso é usado para inserir comentários no seu código (para que se torne mais legível).

Operações aritméticas: Exemplo

Para executar o *script* clique em "Source". Os comandos são executados **na ondem que aparecem** no arquivo.

Note que neste exemplo, nada aparece na tela. Precisamos solicitar que estes valores sejam exibidos na tela com o comando **cat**.

Exemplo:

```
cat("Area: ", area,"\n")
cat("Perimetro",perimetro,"\n")
cat("Diagonal: ", diagonal,"\n")
```

Os valores a serem impressos são concatenados com a vírgula.

A entrada "\n" indica que desejamos iniciar a próxima impressão na próxima linha.

Operações aritméticas: Exemplo

Exemplo completo:

```r
#Atribuindo valor para o lado do quadrado
x <- 2

#Calculando a área
area <- x^2

#Calculando o perímetro
perimetro <- 4*x

#Calculando a diagonal
diagonal <- x*sqrt(2)

#Imprimindo o resultado
cat("Area: ", area,"\n")
cat("Perimetro",perimetro,"\n")
cat("Diagonal: ", diagonal,"\n")
```

Solicitando Informação

Suponha que ao invés de deixar o nosso *script* com o valor de x fixo em 2, desejamos perguntar o valor de x.

Nesse caso podemos usar o comando **readline**. Da seguinte forma:

```
x <- readline("Digite o valor de x: ")
```

Esse comando vai exibir o texto passado como argumento e esperar a resposta do usuário na linha de comando do R.

Solicitando Informação do Usuário

Mas atenção, o comando **readline** retorna uma variável do tipo **character**, mesmo que tenhamos digitado um número.

Portanto, para converter o valor para um valor numérico, podemos usar o seguinte:

```
x <- as.double(readline("Digite o valor de x: "))
```

A função **as.double** converte uma variável de outro tipo em numérico.

Solicitando Informação do Usuário

Mas atenção, o comando **readline** retorna uma variável do tipo **character**, mesmo que tenhamos digitado um número.

Portanto, para converter o valor para um valor numérico, podemos usar o seguinte:

```
x <- as.double(readline("Digite o valor de x: "))
```

A função **as.double** converte uma variável de outro tipo em numérico.

Existem também as funções:

1. **as.integer**: converte uma variável para o tipo inteiro
2. **as.logical**: converte uma variável para o tipo lógico

Ordem de operações

Na matemática, os parênteses destacam a prioridade de cálculo: as operações dentro de parênteses são resolvidas primeiro. Podem ser usados vários tipos de parênteses, como:

- Parênteses ( )
- Colchetes [ ], ou
- Chaves { },

mas estes servem apenas para uma melhor visualização dos pares e não têm influência na ordem.

Em linguagens de programação: **apenas parênteses**.

Ordem de operações

Exemplos:

```
4+2 - 6+10
```

```
## [1] 10
```

```
(4+2) - (6+10)
```

```
## [1] -10
```

Ordem de operações

Qual seria a resposta? *Apenas pense na resposta, não use o R.*

```
2+3*4-5
```

Ordem de operações

Qual seria a resposta? *Apenas pense na resposta, não use o R.*

```
2+3*4-5
```

Sabemos que a multiplicação possui precedência:

```
2+3*4-5
```

```
## [1] 9
```

Ordem de operações

E neste caso, qual seria a resposta?

```
1*(2*(3+4))
```

Ordem de operações

E neste caso, qual seria a resposta?

```
1*(2*(3+4))
```

A expressão dentro do parêntesis sempre possui maior precedência:

```
1*(2*(3+4))
```

```
## [1] 14
```

Ordem de operações

Operação	Descrição
$	seleção
^	potência
-, +	operadores unário para sinal
*, /	multiplicação, divisão
+, -	soma, subtração
<, >, <=, >=, ==, !=	ordenação e comparação
!	negação
&	lógico e
|	lógico ou
<-	atribuição
=	atribuição

Precedência dos operadores, da maior para a menor.

Vetores

Vetores são muito úteis para armazenas vários valores consecutivos. De fato, já vimos vetores quando fazemos gráficos, quando trabalhamos com tabelas.

Vetores somente podem armazenar dados do **mesmo tipo**.

Existem várias maneiras de criar um vetor, a forma mais simples é com o comando **c()**.

Vetores: tipos

Um vetor de inteiros (*integer*) de tamanho 4:

```
vInt <- c( 30L, 40L, 9L, 10L)
```

Um vetor com 3 elementos numéricos (*numeric*, *double*):

```
vDbl <- c( 1, 6.19, 3)
```

Um vetor de texto (*character*) com 3 elementos:

```
vTxt <- c( "Ana", "João", "Maria")
```

Um vetor lógico (*logical*) com 5 elementos:

```
vLog <- c( TRUE, TRUE, FALSE, FALSE, FALSE)
```

Vetores: tipos

Podemos usar o comando **class** para verificar o tipo do vetor, da mesma forma que fizemos antes:

```
class(vInt)
```

```
## [1] "integer"
```

```
class(vDbl)
```

```
## [1] "numeric"
```

```
class(vTxt)
```

```
## [1] "character"
```

```
class(vLog)
```

```
## [1] "logical"
```

Vetores: Criação

Existem outras formas de criar vetores. Já vimos o comando **seq**, que gera vetores iniciando com um valor e incrementando até o final de um intervalo.

```
seq(0,10,0.5)
```

```
##  [1]  0.0  0.5  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0
## [12]  5.5  6.0  6.5  7.0  7.5  8.0  8.5  9.0  9.5 10.0
```

Podemos usar também o operador "**:**" para gerar valores inteiros em um intervalo. No exemplo abaixo, geramos números de 0 à 10.

```
0:10
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10
```

Neste caso, o vetor resultante é de **inteiros**.

Vetores: Criação

Outra forma de gerar vetores no R é através do comando **rep**, que *repete* um valor várias vezes para gerar um vetor:

```
rep(1,10)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
```

Neste exemplo, repetimos o número 1 dez vezes para gerar um vetor de tamanho 10. Abaixo segue outro exemplo, em que repetimos o texto "Olá" 5 vezes:

```
rep("Olá",5)
```

```
## [1] "Olá" "Olá" "Olá" "Olá" "Olá"
```

Vetores: Operações

Operações de multiplicação, divisão, soma, subtração e potência (i.e., *, /, +, -, ^) são todas efetuadas **elemento a elemento** em vetores.

Exemplo

```
x <- c(1,2,3,4)
y <- c(3,3,4,4)
x*y
```

```
## [1]  3  6 12 16
```

Exemplo: Números de Fibonacci

Os números de Fibonacci são os números em uma sequencia de números inteiros cujos dois primeiros números são 0 e 1 e os demais são a soma dos dois anteriores:

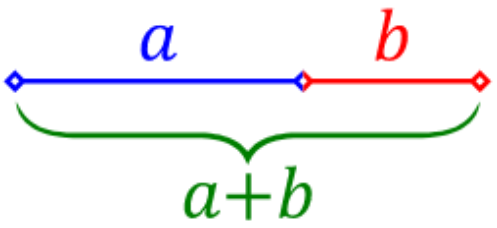
	0	1	2	3	4	5	6	7	8	9	10	11	12
F_n:	0	1	1	2	3	5	8	13	21	34	55	89	144

Assista o vídeo: https://youtu.be/SjSHVDfXHQ4

Exemplo: Números de Fibonacci

Os números de Fibonacci têm conexão com a "Razão Áurea", que é definida como sendo a razão $\varphi = \frac{a}{b}$, para números a e b que satisfazem:

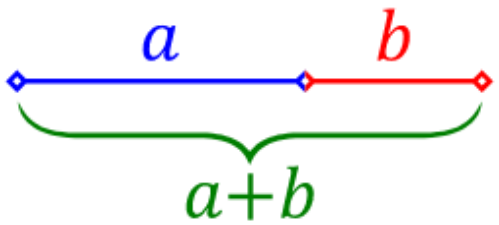
$$\frac{a}{b} = \frac{a+b}{a}$$



Exemplo: Números de Fibonacci

Os números de Fibonacci têm conexão com a "Razão Áurea", que é definida como sendo a razão $\varphi = \frac{a}{b}$, para números a e b que satisfazem:

$$\frac{a}{b} = \frac{a+b}{a}$$



É possível mostrar que:

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.618034\ldots,$$

este número aparece frequentemente na matemática. E possui relação com o que pessoas consideram bonito: aparecem em música, arquitetura, arte...

Exemplo: Números de Fibonacci

Note que a razão entre dois números consecutivos da sequencia de Fibonacci, aproxima a razão aurea:

$$\frac{F_7}{F_6} = \frac{13}{8} = 1.625$$
$$\frac{F_8}{F_7} = \frac{21}{13} = 1.615385$$
$$\frac{F_9}{F_8} = \frac{34}{21} = 1.619048$$
$$\frac{F_{10}}{F_9} = \frac{55}{34} = 1.617647$$

Exemplo: Números de Fibonacci

De fato, podemos mostrar que o n-ésimo número de Fibonacci é dado pela seguinte expressão:

$$F_n = \left\lfloor \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}} \right\rfloor$$

A operação que se assemelha a um colchete significa que o número é arredondado para baixo (chamada de *floor*).

Exemplo: Números de Fibonacci

De fato, podemos mostrar que o n-ésimo número de Fibonacci é dado pela seguinte expressão:

$$F_n = \left\lfloor \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}} \right\rfloor$$

A operação que se assemelha a um colchete significa que o número é arredondado para baixo (chamada de *floor*).

Exercício: Vamos fazer um script em R para fazer o cálculo dos números de Fibonacci usando a expressão acima.

Exemplo: Números de Fibonacci

```r
# Termo da sequência a ser calculado
n <- 10

# Razão áurea
phi <- (1+sqrt(5))/2

# Expressão
Fn <- floor((phi^n - (-phi)^(-n))/sqrt(5))

# Imprimindo o resultado
cat("Fn = ", Fn)
```

```
## Fn =  55
```

Exemplo: Números de Fibonacci

```r
# Termo da sequência a ser calculado
n <- 10

# Razão áurea
phi <- (1+sqrt(5))/2

# Expressão
Fn <- floor((phi^n - (-phi)^(-n))/sqrt(5))

# Imprimindo o resultado
cat("Fn = ", Fn)
```

```
## Fn =  55
```

O que acontece se fizermos $n \leftarrow 0:12$?

Exemplo: Números de Fibonacci

```r
# Termo da sequência a ser calculado
n <- 0:12

# Razão áurea
phi <- (1+sqrt(5))/2

# Expressão
Fn <- floor((phi^n - (-phi)^(-n))/sqrt(5))

# Imprimindo o resultado
cat("Fn = ", Fn)
```

```
## Fn =  0 1 1 2 3 5 8 13 21 34 55 89 144
```

ATIVIDADE EM SALA

Exercício 1

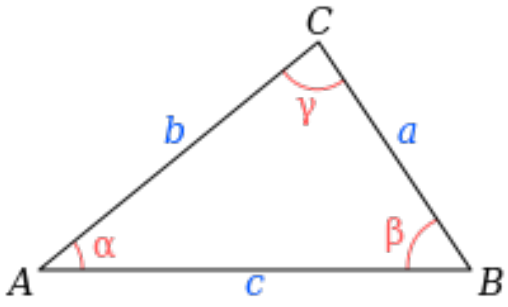
1. Crie um *script* em R que permita a conversão de temperatura de graus Fahrenheit para graus Celsius, usando a fórmula abaixo:

$$C = (F - 32) \times \frac{5}{9}$$

2. Crie outro *script* que permita a conversão de temperatura de graus Celsius para graus Fahrenheit (descubra qual seria a fórmula para conversão usando a expressão acima).

Exercício 2

A fórmula de Heron permite calcular a área de triângulos conhecendo apenas o comprimento de seus lados. Para um triângulo com lados a, b, e c, a fórmula é dada pela seguinte expressão:



$$p = \frac{a+b+c}{2}$$

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

Faça um programa para calcular a área de triângulos usando a fórmula de Heron.

- Leia do teclado 3 números reais, representando os lados do triângulo;
- Calcule e imprima na tela a área do triângulo.

Exercício 3

Faça um programa que:

- Leia do teclado 5 números inteiros maiores do que zero, representando cada um a idade de um indivíduo;
- Calcule e escreva: média, mediana, desvio padrão e variância destas 5 idades.

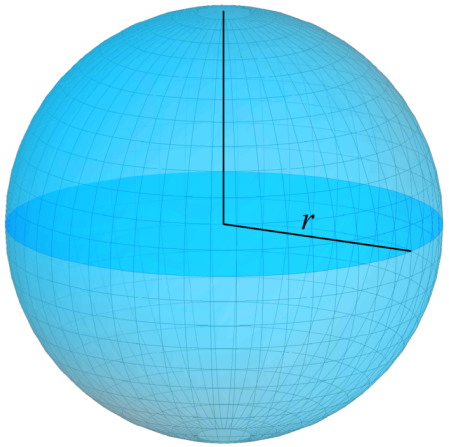
Exercício 4

Faça um programa que lê o valor do raio de uma esfera e calcula o seu volume e sua área. A área da esfera de raio r é dada por:

$$A = 4\pi r^2$$

E o volume de uma esfera de raio r é dado por:

$$V = \frac{4}{3}\pi r^3$$



Referências

- Aulas dos Profs. David Correa Martins Jr, Wagner Tanaka Botelho, Jesús P. Mena-Chalco.
- Livro Bases Computacionais da Ciência.