

MAC0113 - Introdução à Computação para Ciências Humanas

Aula 11 - 1

Sejam bem-vindas, sejam bem-vindos!

**Entre no link <https://app.sli.do/event/pfjoqhgr> ou
e responda a primeira pergunta da aula.**



R. Hirata Jr.

MAC0113 - Introdução à Computação para Ciências Humanas

Aula 11 - 2

Sejam bem-vindas, sejam bem-vindos!

**Entre no link <https://app.sli.do/event/kdtb2zez> ou
e responda a primeira pergunta da aula.**



R. Hirata Jr.

Objetivos de hoje

- Ao final da aula de hoje você deve saber:
 - Comandos de seleção e como usá-los
 - O que é um vetor e como usá-lo

Pequena recordação da aula passada

Expressões aritméticas

Operadores são símbolos especiais (palavras reservadas) que representam funções matemáticas como

- adição (+) e subtração (-),
- multiplicação (*) e divisão (/),
- divisão inteira (%/%)
- exponenciação (**) e
- resto da divisão inteira (%%).

Expressões lógicas

Operadores lógicos também são símbolos especiais (palavras reservadas) que representam funções matemáticas como

- ou lógico (\vee): $0 \vee 0$ - FALSE; $1 \vee 0$, $0 \vee 1$, $1 \vee 1$ - TRUE,
- e lógico (\wedge): $0 \wedge 0$, $1 \wedge 0$, $0 \wedge 1$ - FALSE; $1 \wedge 1$ - TRUE,
- negação (!) - $!1$ - FALSE, $!0$ - TRUE
- comparações: $<$, $>$, \leq , \geq , $==$, $!=$
- ou no comando if $\vee\vee$
- e no comando if $\wedge\wedge$

O comando if

- Estrutura do comando:

```
if (<condição 1>) {  
    comandos a serem executados se a condição 1 for verdadeira  
} else if (<condição 2>) {  
    comandos a serem executados se a condição 2 for verdadeira  
} else if (<condição 3>) {  
    ...  
} else {  
    comandos a serem executados se as condições anteriores forem falsas  
}
```

- Zero, ou mais `else if`, `else` é opcional

O comando if

- Exemplo do uso do comando if

```
a <- as.numeric(readline("Digite um número inteiro: "))
cat("O número digitado foi: ",a,"\n")
b <- as.numeric(readline("Digite um número inteiro: "))
cat("O número digitado foi: ",b,"\n")
if (a < b) {
  cat("O número ",b," é maior que o número ",a,"\n")
} else if (a > b) {
  cat("O número ",a," é maior que o número ",b,"\n")
} else {
  cat("Os números digitados são iguais!")
}
```


Problema em R

- Dada uma sequência de números inteiros positivos, ou zero, imprima a sua soma, o número de **números somados e a média**. A sequência é terminada com um número negativo.

```
SOMA <- 0
CONTA <- 0
MEDIA <- 0
NUM <- as.numeric(readline("Digite um número inteiro: "))
cat("O número digitado foi: ",NUM,"\n")
while (NUM >= 0){
  SOMA <- SOMA + NUM
  CONTA <- CONTA + 1
  NUM <- as.numeric(readline("Digite um número inteiro: "))
  cat("O número digitado foi: ",NUM,"\n")
}
cat("A soma dos números positivos é igual a: ",SOMA,"\n")
cat("A quantidade de números positivos é igual a: ",CONTA,"\n")
if (CONTA != 0) {
  MEDIA = SOMA/CONTA
}
cat("A média dos números positivos é: ",MEDIA,"\n")
```

Desafio em R

- Responda agora no Sli.do

```
# Falso, ou verdadeiro?  
t <- 30 ;  
c <- 25 ;  
resposta <- (t != c) | (c == t%/%2) ;  
print(resposta)
```

```
# Falso, ou verdadeiro?  
x <- (2 == 3) ;  
y <- (2 != 3) ;  
z <- !(x == y) ;  
print(z)
```

Vetores (arrays)

- Um vetor é uma sequência de objetos todos do mesmo tipo
- Há três tipos primitivos em R:
 - numérico (numeric) - números
 - alfanumérico (character) - caracteres
 - lógico (logical) - {verdadeiro, falso} - {true, false}
- Pode-se criar vetores de quaisquer tipos primitivos do R
- Uma das formas para se criar um vetor em R é usar o operador de concatenação:
 - `c`
 - Por exemplo
 - `reportCovid = c(4090,2067,530,499,400,364,322,303,298,258,236,200,188)`
 - `newCases =`
`c(310822,69300,18873,25642,21469,8790,6796,5547,2897,2880,11472,7723,`
`9981)`

Vetores (arrays)

- Uma vez criado, podemos manipular todos os elementos do vetor usando apenas um nome para quantos valores há no vetor.
- Para isso, precisamos usar o índice do vetor que começa com 1 e termina com o número de elementos do vetor. Além disso, precisamos usar os colchetes ao redor do índice: [<índice>] (os sinais de <> são apenas para indicar o que vai dentro dos colchetes).
- Por exemplo:
- `reportCovid = c(4090,2067,530,499,400,364,322,303,298,258,236,200,188)`
- `print(reportCovid[1])`
- 4090
- `print(reportCovid[13])`
- 188
- `V = c()`
- `V = c(V,1)`

Vetores (arrays)

Para imprimir o vetor inteiro e seus índices, podemos fazer:

```
reportCovid <- c(4090,2067,530,499,400,364,322,303,298,258,236,200,188)
reportCovidn <- length(reportCovid)
print(reportCovidn)
print("O vetor reportCovid contém os elementos:")
i = 1
while (i <= reportCovidn) {
  cat(i,reportCovid[i],"\n")
  i = i + 1
}
```

Pronto, pode acordar! Agora é pra valer!

Faça todos os exercícios abaixo

1. Dada uma sequência de números inteiros positivos, ou zero, terminada com um número negativo, armazene-a em um vetor. Após o usuário ter finalizado a digitação da sequência, imprima o vetor.
2. Dada uma sequência de números inteiros positivos, ou zero, terminada com um número negativo, armazene-a em um vetor e, usando o vetor, imprima a soma dos números pares e a soma dos números ímpares separadamente.
Dica: para saber se o número é par, verifique o se o resto da divisão dele por dois é zero. O resto da divisão é dado pelo operador `%%`. Por exemplo $5\%2 = 1$. Curiosidade, para obter o quociente da divisão, use `% / %`. Por exemplo $5\%/%2 = 2$.

Problema em R

● Dada uma sequência de números inteiros positivos, ou zero, terminada com um número negativo, armazene-a em um vetor. Após o usuário ter finalizado a digitação da sequência, imprima o vetor no formato: índice - elemento.

```
numeros = c()

num <- as.numeric(readline("Digite um número inteiro, por favor. Termine o programa com um número 0: "))
print(num)

while (num >= 0) {
  numeros = c(numeros, num)
  num <- as.numeric(readline("Digite um número inteiro, por favor. Negativo para terminar: "))
  print(num)
}

tamanhoNumeros = length(numeros)
i = 1
while (i <= tamanhoNumeros) {
  cat(i, numeros[i], "\n")
  i = i+1
}
```


Problema em R

- Dada uma sequência de números inteiros positivos, ou zero, terminada com um número negativo, armazene-a em um vetor e, usando o vetor, imprima a soma dos números pares e a soma dos números ímpares separadamente.

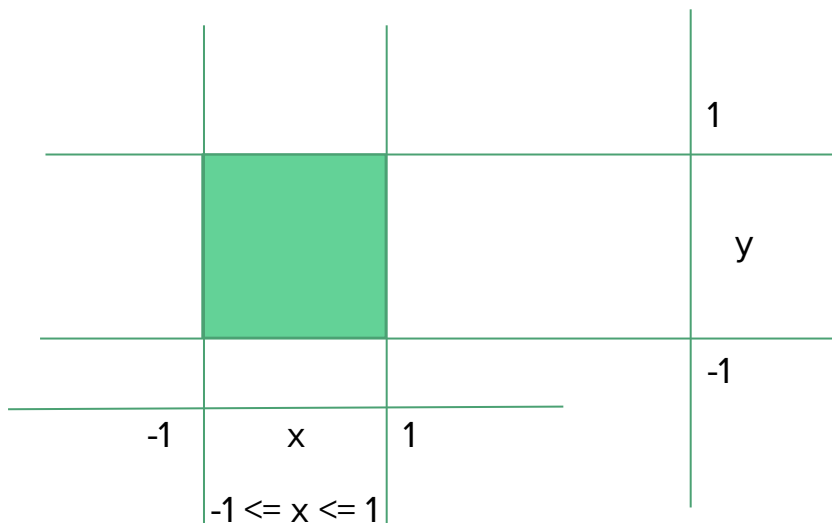
```
somaPar <- 0
somaImpar <- 0
numeros <- c()
num <- as.numeric(readline("Digite um número inteiro, por favor. Termine o programa com um número 0: "))
print(num)
while (num >= 0) {
  numeros = c(numeros,num)
  num <- as.numeric(readline("Digite um número inteiro, por favor. Negativo para terminar: "))
  print(num)
}
tamanhoNumeros = length(numeros)
i = 1
while (i<=tamanhoNumeros) {
  if (numeros[i] %% 2 == 0)
    somaPar <- somaPar + numeros[i]
  else
    somaImpar <- somaImpar + numeros[i]
  i = i+1
}
cat("Os números pares somam ",somaPar," e os números ímpares somam ",somaImpar,"\n")
```

Faça todos os exercícios abaixo

1. Dados dois números quaisquer, correspondentes às coordenadas de um ponto num plano cartesiano, verifique se esse ponto pertence ao quadrado de vértices localizados nas posições $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$. Caso o ponto pertença ao quadrado, imprima a frase “O ponto pertence ao quadrado.”, ou “O ponto não pertence ao quadrado.”, caso contrário.

Problema em R

- Dados dois números quaisquer, correspondentes às coordenadas de um ponto num plano cartesiano, verifique se esse ponto pertence ao quadrado de vértices localizados nas posições $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$. Caso o ponto pertença ao quadrado, imprima a frase “O ponto pertence ao quadrado.”, ou “O ponto não pertence ao quadrado.”, caso contrário.



$$-1 \leq y \leq 1$$

$$(y \geq -1) \&\& (y \leq 1)$$

$$(x \geq -1) \&\& (x \leq 1)$$

$$(x \geq -1) \&\& (x \leq 1) \&\& (y \geq -1) \&\& (y \leq 1)$$

Problema em R

- Dados dois números quaisquer, correspondentes às coordenadas de um ponto num plano cartesiano, verifique se esse ponto pertence ao quadrado de vértices localizados nas posições $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$. Caso o ponto pertença ao quadrado, imprima a frase “O ponto pertence ao quadrado.”, ou “O ponto não pertence ao quadrado.”, caso contrário.

```
x <- as.numeric(readline("Digite a coordenada X do ponto desejado, por favor: "))
y <- as.numeric(readline("Digite a coordenada Y do ponto desejado, por favor: "))
cat("O ponto digitado foi: (",x,"",",",y,"")\n")

if ((x >= -1) && (x <= 1) && (y >= -1) && (y <= 1)) {
  cat("O ponto digitado, (",x,"",",",y,""), pertence ao quadrado\n")
} else {
  cat("O ponto digitado, (",x,"",",",y,""), não pertence ao quadrado\n")
}
```

Problema em R

Dados dois números quaisquer, correspondentes às coordenadas de um ponto num plano cartesiano, verifique se esse ponto pertence ao quadrado de vértices localizados nas posições $\{(-1,-1),(-1,1),(1,-1),(1,1)\}$. Caso o ponto pertença ao quadrado, imprima a frase “O ponto pertence ao quadrado.”, ou “O ponto não pertence ao quadrado.”, caso contrário.

```
x <- as.numeric(readline("Digite a coordenada X do ponto desejado, por favor: "))
y <- as.numeric(readline("Digite a coordenada Y do ponto desejado, por favor: "))
cat("O ponto digitado foi: (",x,"","y,")\n")

pertenceQuadrado <- FALSE

if (x >= -1)
  if (x <= 1)
    if (y >= -1)
      if (y <= 1) {
        cat("O ponto digitado, (",x,"","y,") , pertence ao quadrado\n")
        pertenceQuadrado <- TRUE
      }

if (!pertenceQuadrado) {
  cat("O ponto digitado, (",x,"","y,") , não pertence ao quadrado\n")
}
```

Faça todos os exercícios abaixo

1. Dado um número inteiro positivo, que representa um ano, verifique se ele é um ano bissexto. Se for, imprima a mensagem: “O ano digitado é bissexto”. Se não for, imprima “O ano digitado não é bissexto”.
2. Dado um vetor com uma sequência de números positivos quaisquer, determine se a sequência está em ordem crescente.
3. Dado um vetor com uma sequência com as notas finais de MAC113, determinar quantos estudantes:
 - foram aprovados: nota final ≥ 5 ;
 - estão de recuperação: $3 \leq \text{nota final} < 5$;
 - foram reprovados: nota final < 3 .

Um ano é bissexto se ele é múltiplo de 4 e não é múltiplo de 100, ou se ele é múltiplo de 400. Por exemplo, os anos 1700, 1800, 1900 não são anos bissextos. Os anos 1600, 2000 e 2020 o são.

Obrigado!
