

## Lista de Exercícios II

### MAP2112 – Introdução a Lógica de Programação e Modelagem Computacional

Prof. Dr. Luis Carlos de Castro Santos

1º semestre 2021

#### Seção 1 – Laços e Condicionais

##### Exercício 1.1

Crie um script em R que calcula a raiz quadrada de um vetor inteiro de comprimento 1, se o valor contido no vetor for negativo o script deve retornar NA

##### Exercício 1.2

Crie um script em R que retorna TRUE se os elementos de um vetor  $x$  inteiro de comprimento 3 são estritamente crescentes.

##### Exercício 1.3

Crie um script em R que retorna a quantidade valores que são maiores que a média de um vetor. Você pode usar o comando *mean()* e deve obrigatoriamente usar o comando *for* (Dica: crie um contador)

#### Seção 2 – Laços – parte 1

##### Exercício 2.1

Dada a variável *msg* <- c("Hello") e iniciando a contagem com *i* <- 1

Crie um laço com o comando repeat{} termina quando *i* atinge o valor 12 e escreve a variável *msg* para os valores de *i* múltiplos de 3. (Dica: O comando *x%%y* é o resto da divisão *x/y*)

##### Exercício 2.2

Aplicando o comando

```
n <- sample(1:6,1)
```

gera-se um valor *n* aleatório inteiro entre 1 e 6. A variável *n* simula o lançamento de um dado.

Crie um laço com o comando while{} que termina após 3 sorteios de números ímpares.

##### Exercício 2.3

Aplicando o comando

```
alfabeto <- letters[sample(1:26,26)]
```

gera-se um vetor de ordem aleatória de todas letras do alfabeto romano (26 letras), ou seja um alfabeto embaralhado.

Crie um script em R usando um laço de `for{}` que encontra em que posição do alfabeto embaralhado está a letra “a” .

#### **Exercício 2.4**

Aplicando o comando

```
n <- sample(1:10,1)
```

gera-se um valor n aleatório inteiro entre 1 e 10.

Crie um script com um laço de `while{}` onde uma variável i evolui de 1 a 10 e use o comando `break` para quebrar o laço quando i for igual ao valor aleatório.

#### **Exercício 2.5**

Usando novamente o comando

```
n <- sample(1:10,1)
```

gera-se um valor n aleatório inteiro entre 1 e 10.

Crie um script com um laço de `while{}` onde uma variável i evolui de 1 a 10, imprima o valor de dessa variável e use o comando `next` para pular o valor em que i é igual ao valor aleatório.

### **Seção 3 – Laços – parte 2**

#### **Exercício 3.1**

Crie um script que imprime uma variável aleatória `x <- rnorm(1)` e que só termina quando x é maior que 1

#### **Exercício 3.2**

Adapte o script anterior para não imprimir os números negativos.

#### **Exercício 3.3**

Adapte o script anterior para contar quantos números negativos não foram impressos

#### **Exercício 3.4**

Usando os comandos apresentados para simular o lançamento de uma moeda por 20 vezes, sendo cara representada por 1 e coroa por 0 e armazene os resultados da simulação num vetor.

#### **Exercício 3.5**

Crie um script para usando laços aninhados (um laço de `for{}` dentro de outro laço de `for{}`, por exemplo) para gerar matrizes de Hilbert, conforme o exemplo para dimensão 5. (Dica: qual a lei de formação dos coeficientes em função dos índices ?)

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}.$$

### Exercício 3.6

Crie um script para descobrir o número máximo de termos quando o produto da sequência  $1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots$

Ainda fica abaixo de 10 milhões.

### Exercício 3.7

Crie um script que simula aleatoriamente o valor de uma ação começando em 100 e com variação aleatória normalmente distribuída de media 0 e desvio padrão 0.01 a cada período (  $rnorm(1,0, 0.01)$  ). Apresente o tempo para que o valor da ação caia abaixo de 50 ou passe de 150. (Bonus: Apresente os resultados num gráfico)

## Seção 4 – Funções

### Exercício 4.1

Crie uma função que retorna TRUE se um dado inteiro  $v$  pertence a um dado vetor  $x$ . Faça um script que testa se a função funciona em todos os casos esperados.

### Exercício 4.2

Crie uma função que dado um inteiro e um vetor retorna quantos vezes o interior aparece e em que posições. Faça um script que testa se a função funciona.

### Exercício 4.3

Crie uma função que dado um inteiro calcula quantos divisores ele tem (além de 1 e si próprio). Imprima os divisores enquanto conta.