

## LeanOn Systems - Prova de Seleção

Nome : Hudson Lyrio Cratz

Assinatura: Hudson Lyrio Cratz

Data: 03/07/2021

Hora Início: 08:00

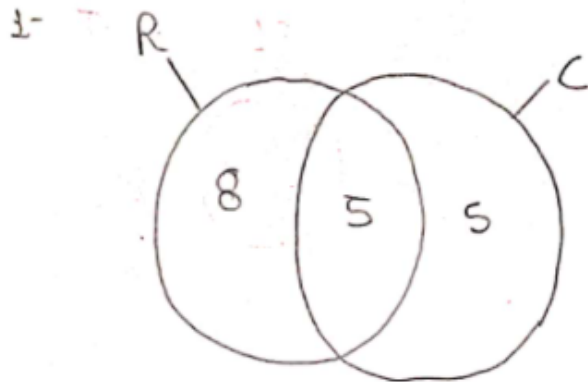
Hora Fim: 12:00

### Parte 1 – Matemática e Lógica

Instruções:

- Utilizar caneta ou lápis.
- Utilizar folhas de rascunho onde for necessário. O rascunho deverá ser entregue junto com a prova.

**1** - Todos os amigos presentes numa festa tomam refrigerante ou cerveja. Treze amigos tomam refrigerante, dez tomam cerveja e 5 tomam refrigerante e cerveja. Quantos amigos estão presentes na festa?



$$\text{Total} = 8 + 5 + 5 = 18 \text{ amigos}$$

Resposta: 18 Amigos

**2** - Sarah supõe que seu relógio está 5 minutos atrasado, mas, na verdade, ele está 10 minutos adiantado. Sarah chega para um encontro marcado, julgando estar 15 minutos atrasada em relação ao horário combinado. Na realidade ela chegou: na hora, atrasada ou adiantada por quantos minutos?

2-

- 15 min que Sarah julga está atrasada
- 5 min que Sarah supõe que o relógio está atrasado
- 10 min que o relógio está adiantado

$$X = 15 - 5 - 10 = 0$$

Na hora!

Resposta: Na Hora

3 - Uma pizzaria realiza uma promoção com o anúncio "Compre uma e leve outra pela metade do preço". Outra promoção que a pizzaria poderia fazer oferecendo o mesmo desconto percentual é:

- (a) "Leve duas e pague uma"
- (b) "Leve três e pague uma"
- (c) "Leve três e pague duas"
- (d) "Leve quatro e pague três"**
- (e) "Leve cinco e pague quatro"

3 - Piza      Valor

|   |   |     |
|---|---|-----|
| 2 | — | 1,5 |
| 4 | — | 3   |

(D)

4 - Dos oito algarismos do número do telefone de Lorena, Sarah se lembra apenas dos quatro primeiros na ordem correta, mas apesar de se lembrar dos quatro últimos algarismos e saber que nenhum deles se repete, esqueceu-se da ordem destes. Qual é o maior número de tentativas que Sarah pode fazer até conseguir acertar o número do telefone de Lorena?

4 -

$$\text{— — — — — } \underline{4} \times \underline{3} \times \underline{2} \times \underline{1} = 24 \text{ Possibilidades}$$

Resposta: 24 Possibilidades

## Parte 2 – Programação

Instruções:

- Utilizar caneta ou lápis.
- Você pode resolver as questões em pseudocódigo ou na linguagem de sua preferência, indicando esta linguagem.
- Sinta-se à vontade para escrever observações, se achar necessário.

**1** – Uma palavra é dita palíndroma se ela é simétrica, isto é, se a escrevermos de trás para frente, o resultado é o mesmo. Por exemplo, "ARARA" é palíndroma, mas "ARAR" não é. Escreva um pequeno programa, que determina se uma palavra é palíndroma.

**2** – Uma imensa lista telefônica contendo pares da forma {número do telefone, nome da pessoa} foi armazenada como um vetor ordenado pelo nome em ordem alfabética. Escreva um programa que encontra o número do telefone de uma dada pessoa nessa lista, levando em conta que a lista é muito grande e que os usuários necessitam que o resultado da busca seja o mais rápido possível.

**3** – Considere o seguinte esquema de banco de dados:

| TABELA       | COLUNAS                                      |
|--------------|--|
| FORNECEDOR   | CODIGO_FORNECEDOR, NOME_FORNECEDOR, CIDADE   |
| PEÇA         | CODIGO_PEÇA, NOME_PEÇA, PREÇO                |
| CARRO        | CODIGO_CARRO, NOME_CARRO, TIPO               |
| FORNECIMENTO | CODIGO_FORNECEDOR, CODIGO_PEÇA, CODIGO_CARRO |

Escreva um comando SQL que seja capaz de consultar os fornecedores localizados na cidade de nome "VITORIA" que fornecem a peça de código "MOTOR" para o carro de código "KOMBI", com os respectivos preços.

Exemplo:

| FORNECEDOR | PREÇO |
|------------|-------|
| Superpeças | 1.000 |
| Peçauto    | 1.500 |

**4** – Seu amigo está desenvolvendo um pequeno programa de processamento de imagens e pediu sua ajuda para implementar a funcionalidade semelhante ao "balde de tinta" do MS-Paint.

O programa de seu amigo representa imagens utilizando matrizes de caracteres, sendo que cada valor da matriz representa um pixel e letras e símbolos representam as diferentes cores.

Por exemplo, a seguinte matriz 4x6 representa a letra P na cor "#", com cor de fundo "." (ponto)

```
.###..
.#..#.
.###..
.#....
```

Sua subrotina deve receber um pixel e uma nova cor e pintar a região desse pixel com a nova cor, como a ferramenta "balde de tinta" do MS-Paint.

Exemplos:

| Pixel (0,1) e nova cor 'o' |        | Pixel (1,3) e nova cor 'o' |        | Pixel (1,3) e nova cor '#' |        |
|----------------------------|--------|----------------------------|--------|----------------------------|--------|
| Antes                      | Depois | Antes                      | Depois | Antes                      | Depois |
| .###..                     | .ooo.. | .###.                      | .###.. | .###..                     | .###.. |
| .#..#.                     | .o..#. | .#..#.                     | .#oo#. | .#..#.                     | .####. |
| .###..                     | .ooo.. | .###.                      | .###.. | .###..                     | .###.. |
| .#....                     | .o.... | .#....                     | .#.... | .#....                     | .#.... |

Obs: Fiz as questões com arquivos .js e .sql

### Questão 1

```
JS palindrome.js > ...
1  // function to check if the string is a palindrome
2  function palindrome(string) {
3
4      for(let i=0; i< string.length/2; i++){
5          if(string[i] !== string[string.length -1 -i]) return `${string} is not a Palindrome`;
6      }
7      return `${string} is a palindrome`
8  }
9
10
11 console.log(palindrome('arara'))
```

palindrome.js

### Questão 2

```
JS phoneList.js > generateName
1  // function to capitalize the first letter
2  function capFirst(string) {
3      return string.charAt(0).toUpperCase() + string.slice(1);
4  }
5  // function to generate random number
6  function getRandomInt(min, max) {
7      return Math.floor(Math.random() * (max - min)) + min;
8  }
9  // function to generate a random Name (test)
10 > function generateName(){...
18 }
```

```
// function to genate a list of fake users
function fakeUsers() {
    var list = [], user;

    for(i=0; i<100; i++) {
        user = {
            'name' : generateName(),
            'phone' : `+55 ${Math.floor((Math.random() * 90) + 10)} ${Math.floor(Math.random() * 10000)}-${Math.floor(Math.random() * 10000)}`
        }

        list.push(user)
    }

    return list
}

// function to sort the list of fake users
function sortByName(a, b) {
    return (a.name > b.name) ? 1 : ((b.name > a.name) ? -1 : 0);
}
```

```

39
40 // function to find a user at list
41 function search(name) {
42
43     // List generated by function fakeUsers()
44     // this was a trick to generate a static list, down below the command that I used to generate the list
45     // const list = fakeUsers()
46
47 > const list = [ ...
148     ];
149
150     list.sort(sortByName)
151     user = list.find(element => element.name == name)
152     if(user) return user;
153     return 'User not found'
154 }
155 console.log(search('Empty Buddy'));

```

phoneList.js

### Questão 3

```

searchBD.sql
1 SELECT FORNECEDOR.CODIGO_FORNECEDOR, PECA.PRECO FROM FORNECEDOR, PECA, FORNECIMENTO
2 WHERE FORNECEDOR.CODIGO_FORNECEDOR = FORNECIMENTO.CODIGO_FORNECEDOR
3 AND FORNECEDOR.CIDADE = 'VITORIA' AND FORNECIMENTO.CODIGO_Peca = 'MOTOR' AND FORNECIMENTO.CODIGO_CARRO = 'KOMBI'
4 AND PECA.CODIGO_Peca = FORNECIMENTO.CODIGO_Peca

```

searchBD.sql

### Questão 4

```

JS paintBucket.js > ...
1 // function to create the initial design
2 function createDesign() {
3     const matrix = [
4         ['.', '#', '#', '#', '.', '.'],
5         ['.', '#', '.', '.', '#', '.'],
6         ['.', '#', '#', '#', '.', '.'],
7         ['.', '#', '.', '.', '.', '.']
8     ];
9     return matrix;
10 }
11 // function to print design on the screen
12 function printDesign(matrix) {
13     print = '';
14     for(i=0; i<matrix.length; i++) {
15         for(j=0; j<matrix[i].length; j++){
16             print += matrix[i][j] + ' ';
17             if(j === matrix[i].length-1) print += '\n';
18         }
19     }
20     return print;
21 }
22

```

```
22
23 // A recursive function to replace
24 // previous color 'currentCollor' at pixel '(x, y)'
25 // and all surrounding pixels of (x, y)
26 // with new color 'newColor'
27 function changeColor(matrix, x, y, currentCollor, newColor)
28 {
29     // Base cases
30     if (x < 0 || x >= matrix.length || y < 0 || y >= matrix[0].length)
31         return;
32     if (matrix[x][y] != currentCollor)
33         return;
34     if (matrix[x][y] != currentCollor)
35         return;
36
37     // Replace the color at cell (x, y)
38     matrix[x][y] = newColor;
39
40     // Recursively call for rigth, left, up and down
41     changeColor(matrix, x+1, y, currentCollor, newColor);
42     changeColor(matrix, x-1, y, currentCollor, newColor);
43     changeColor(matrix, x, y+1, currentCollor, newColor);
44     changeColor(matrix, x, y-1, currentCollor, newColor);
45 }
46
```

```
47 // paintBucket finds the previous color on (x, y) and
48 // calls the function changeColor
49 function paintBucket(pixel, newColor)
50 {
51     matrix = createDesign();
52     x = pixel.x;
53     y = pixel.y;
54     currentCollor = matrix[x][y];
55     changeColor(matrix, x, y, currentCollor, newColor);
56 }
57
58
59 matrix = createDesign();
60
61 //print initial design
62 console.log(printDesign(matrix));
63
64 pixel = {
65     x: 0,
66     y: 1
67 }
68
69 // calling the subroutine
70 paintBucket(pixel, 'o');
71
72 //print updated design
73 console.log(printDesign(matrix));
74
```

**paintBucket.js**