

PARA AS QUESTÕES DE PROGRAMAÇÃO FOI UTILIZADO PYTHON.
PARA AS QUESTÕES DE BANCO DE DADOS, UTILIZEI MySQL

1 - Escreva uma função que receba um valor inteiro como parâmetro de entrada e imprima na tela n linhas conforme estrutura apresentada abaixo. Por exemplo, as seguintes linhas devem ser apresentadas se o parâmetro de entrada for 10.

NOTA: Caso seja inserido um valor menor ou igual a zero, uma crítica deverá ser exibida e o processo deverá ser abortado.

PYTHON

```
valor = int(input("Informe um número: "))

if valor > 0:
    for c in range(valor, 0, -1):
        print("\n", end="")
        for valor in range(c, 0, -1):
            print(f"{valor*valor} ", end="")
else:
    print("São aceitos apenas valores maiores que 0")
```

2 - Escreva uma função que receba um parâmetro do tipo string e retorne uma string como resultado. A função deverá "compactar" a string recebida como parâmetro de entrada. A compactação funcionará escrevendo o caractere encontrado seguido da quantidade de vezes que ele ocorre em sequência.

Ex.:

Parâmetro de entrada: jjjjooao

Resultado da função: j4o2ao2

```
def compacta_string(s: string):
    if not s:
        return ""

    resultado = ""
    contador = 1
    for i in range(1, len(s)):
        if s[i] == s[i - 1]:
            contador += 1
        else:
            resultado += s[i - 1] + str(contador)
            contador = 1
    resultado += s[-1] + str(contador)
```

```

    return resultado

entrada = input("Digite algo: ")
print(compacta_string(entrada)) # Saída: j4o2a1o2

```

3 - Escreva um programa que leia números positivos do teclado, até que o número zero seja digitado. Após, o programa deverá exibir um relatório na tela descrevendo os seguintes itens:

- a) Quantos números foram lidos.
- b) O maior numero lido.
- c) A média dos números lidos.
- d) O menor número ímpar lido (caso algum número ímpar tenha sido digitado).
- e) A quantidade de vezes que cada número ocorreu. Exemplo: "O número 7 ocorreu 2 vezes." "O número 13 ocorreu 8 vezes".

DICA: Use vetores.

```

numero = 1
numerosLidos = []
quantidadeNumerosLidos = 0
soma = 0
media = 0
numerosImpares = []
quantidadeAparece = []

vistoQuantidade = {}
vistoLidos = set()

while True:
    numero = int(input("Digite um número: "))
    if numero == 0:
        break
    quantidadeNumerosLidos += 1
    numerosLidos.append(numero)
for numero in numerosLidos:
    soma += numero
media = soma / quantidadeNumerosLidos

for numero in numerosLidos:
    if numero % 2 == 1:
        numerosImpares.append(numero)

maiorNumeroLido = max(numerosLidos)
menorNumeroImparLido = min(numerosImpares)

```

```

for numero in numerosLidos:
    if numero in vistoQuantidade:
        vistoQuantidade[numero] += 1
    else:
        vistoQuantidade[numero] = 1

print("Quantidade de números lidos:", quantidadeNumerosLidos)
print("Maior número lido:", maiorNumeroLido)
print("Menor número ímpar lido:", menorNumeroImparLido)
print("Média dos números lidos:", media)

for numero, quantidade in vistoQuantidade.items():
    print("O número {} aparece {} vezes".format(numero,
quantidade))

```

4 - 4 - Escreva um programa que leia um arquivo texto (.txt) escolhido pelo usuário. Após a leitura do arquivo, o programa deverá exibir qual linha possui mais vogais e qual linha possui mais consoantes. Por simplicidade admita que o arquivo conterá apenas letras (sem acentos ou ç) e espaços em branco. Caso ocorra empate, qualquer uma das linhas poderá ser exibida.

```

def contar_vogais_consoantes(linha):
    vogais = "aeiouAEIOU"
    consoantes = "bcdfghjklmnpqrstvwxyzBCDEFGHJKLMNPQRSTUVWXYZ"
    contagem_vogais = sum(1 for char in linha if char in vogais)
    contagem_consoantes = sum(1 for char in linha if char in
consoantes)
    return contagem_vogais, contagem_consoantes

def contagem_letras():
    nome_arquivo = input("Digite o nome do arquivo .txt (incluindo
a extensão): ")
    with open(nome_arquivo, 'r') as arquivo:
        linhas = arquivo.readlines()
        max_vogais = 0
        max_consoantes = 0
        linha_max_vogais = ""
        linha_max_consoantes = ""
        for linha in linhas:
            vogais, consoantes = contar_vogais_consoantes(linha)
            if vogais > max_vogais:
                max_vogais = vogais
                linha_max_vogais = linha
            if consoantes > max_consoantes:
                max_consoantes = consoantes

```

```

        linha_max_consoantes = linha

    print("Linha com mais vogais:")
    print(linha_max_vogais.strip())
    print(f"Quantidade de vogais: {max_vogais}")
    print("\nLinha com mais consoantes:")
    print(linha_max_consoantes.strip())
    print(f"Quantidade de consoantes: {max_consoantes}")

contagem_letras()

```

5 - Escreva o script necessário para a criação de uma tabela chamada CLIENTES e outra chamada TELEFONES. A tabela CLIENTES deverá possuir os seguintes campos: NOME, CPF e IDADE sendo o CPF a chave primária.

A tabela TELEFONES deverá possuir três campos, sendo eles: CPF_CLIENTE, DDD e TELEFONE, para esta tabela o campo CPF_CLIENTE deverá ser a chave primária e estrangeira (referenciando o campo CPF da tabela CLIENTES)

MYSQL

```

create database desafio_tasken;
use desafio_tasken;
CREATE TABLE CLIENTES (
    NOME varchar(30),
    IDADE int,
    CPF char(11) PRIMARY KEY
);

CREATE TABLE TELEFONES(
    DDD int,
    TELEFONE int unique,
    CPF_CLIENTE char(11) PRIMARY KEY,
    CPF char(11),
    CONSTRAINT FK_CPF_CLIENTE FOREIGN KEY (CPF) REFERENCES CLIENTES(CPF)
);

```

6 - Escreva uma consulta SQL para obter o nome de todos os clientes que possuam idade igual ou superior a 22 anos. O resultado deverá estar ordenado pela idade de forma crescente.

```

select NOME from CLIENTES
where IDADE >= 22
ORDER BY IDADE ASC;

```

7 - Escreva uma consulta SQL que exiba o nome dos clientes e a quantidade de telefones encontrados. A consulta deverá exibir somente o nome dos clientes que possuam pelo menos 1 telefone.

```
select CLIENTES.NOME, count(TELEFONES.TELEFONE) as quantidade_telefones from  
CLIENTES  
join TELEFONES ON CLIENTES.CPF = TELEFONES.CPF_CLIENTE  
group by CLIENTES.nome  
having count(TELEFONES.TELEFONE) >0;
```

8 – Escreva o comando SQL necessário para excluir todos os clientes que possuam o sobrenome 'santos'.

```
delete from CLIENTES where NOME Like '%SANTOS%'
```