Programação e SQL

1. Identificação

Nome

Endereço de email

Número de telefone

2. **Para as questões que envolvem desenvolvimento de código, procure utilizar a sintaxe C/C++ ou Java para representar o código.**

O resultado deverá ser enviado para uma conta no Github / Gitlab ou Bitbucket.

Utilize um arquivo por questão e adicione um arquivo Readme com comentários de como testar e validar sua solução ou outros comentários que julgar pertinentes como por exemplo qual o motivo de ter utilizado determinada abordagem (questões performáticas, clareza, manutenção do código, etc).

0 de 7 respondidas

de programação, utilize a ferramenta https://onlinegdb.com para testar e validar sua solução. Já para problemas de implementação SQL, utilizar a ferramenta https://sqliteonline.com para testar e validar sua solução.
Ao final, disponibilizar o link da solução testada e validada junto a resposta do problema no repositório.

Finalmente, utilize o campo comentário de cada uma das questões para informar o link do repositório com sua resposta.

Considere o seguinte problema:

Escreva um programa que imprime cada número de 1 até 100 em uma nova linha.

Para cada múltiplo de 3, imprima "Foo", ao invés do número.
Para cada múltiplo de 5, imprima "Baa", ao invés do número.
Para números múltiplos simultaneamente de 3 e 5, imprima "FooBaa", ao invés do número.

A sua solução deverá ser utilizando o menor número de linhas de código possível porém deve produzir um código eficiente.

3. Considere uma string contendo caracteres

0 de 7 respondidas

1. **Concatenar** um caractere minúsculo do alfabeto português ao final da string.
2. **Remover** o último caractere da string. Se a string estiver vazia, ela permanecerá vazia.

Dado um número inteiro ***k*** e duas strings ***s*** e ***t***, determine se você consegue converter ***s*** em ***t*** através de exatamente ***k*** operações descritas acima sobre ***s***. Se possível, o programa imprime '**sim**', do contrário imprime '**não**'.

Por exemplo, string s = [a, b, c] e string t = [d, e, f]. O número de movimentos k = 6. Para converter s em t, primeiro removemos todos os caracteres usando 3 movimentos. Em seguida concatenamos cada um dos caracteres de t na ordem. No sexto movimento, você terá a string s esperada. Se houver mais movimentos disponíveis que o necessário, eles podem ser eliminados executando múltiplas remoções em uma string vazia. Se houver movimentos a menos, não seria possível criar a nova string.

Desenvolva um programa que implementa e utiliza a função **ConcatERemove(s,t,k)**. Ela deve retornar os resultados 'sim' ou 'não'.
A função tem os seguintes parâmetros:
s: string inicial
t: string desejada
k: um número inteiro que representa o número de operações

Formato de entrada
A primeira linha contêm a string s, a string inicial.
A segunda linha contém a string t, a string desejada.
A terceira linha contém um inteiro k, o número

0 de 7 respondidas

Limitações
a) 1 <= |s| <= 100
b) 1 <= |t| <= 100
c) 1 <= k <= 100
d) s e t consiste de letras minúsculas do alfabeto português, ascii[a-z]

Formato de saída
Imprima 'sim' se você puder obter a string t executando exatamente k operações sobre a string s, e imprime 'não' no caso contrário.

Exemplo 1
```
blablablabla
blablabcde
8

sim
```

Explicação
Foram necessários 5 operações para reduzir a string s para "blablab". Em seguida, foram necessários 3 operações de concatenação (c,d,e), para obter a string t "blablabcde". Como foi possível converter s em t utilizando exatamente k=8 operações, o programa imprimiu "sim".

Exemplo 2
```
aba
aba
7

sim
```

Explicação
Foram necessários 4 operações de redução da

string vazia). Em seguida foram executadas 3 operações de concatenação (a,b,a). Como foi possível converter s em t utilizando exatamente k=7 operações, o programa imprimiu "sim".

Exemplo 3

```
ashley
ash
2

não
```

4. Considere o seguinte problema

São fornecidas duas tabelas: Alunos e Notas.

Alunos contém três colunas: ID, Nome e Valor.

Coluna	Tipo
ID	Inteiro
Nome	String
Valor	Inteiro

Notas possui os seguintes dados:

Nota	Valor_Min	Valor_Max
1	0	9
2	10	19
3	20	29
4	30	39
5	40	49

0 de 7 respondidas

7	60	69
8	70	79
9	80	89
10	90	100

Joana dá a Eva a tarefa de gerar um relatório contendo três colunas: Nome, Nota e Valor. Joana não quer os Nomes dos alunos que receberam uma nota inferior a 8. O relatório deve estar em ordem decrescente por nota, ou seja, as notas mais altas são inseridas primeiro. Se houver mais de um aluno com a mesma nota (8-10) atribuído a eles, ordene esses alunos em particular por seus nomes em ordem alfabética. Por fim, se a nota for inferior a 8, use "NULL" como nome e liste-os por notas em ordem decrescente. Se houver mais de um aluno com a mesma nota (1-7) atribuído a eles, ordene esses alunos em particular por suas notas em ordem crescente.

Escreva uma consulta SQL para ajudar Eva.

Exemplo de entrada

ID	Nome	Valor
1	Julia	88
2	Carol	68
3	Maria	99
4	Andreia	78
5	Jaqueline	63
6	Marcela	81

Exemplo de saída

Maria 10 99 Marcela 9 81

0 de 7 respondidas

```
Andreia 8 78
NULL 7 63
NULL 7 68
```

Observação: Imprima "NULL" no nome se a nota for inferior a 8.

Explicação
Considere a seguinte tabela com as notas atribuídas aos alunos:

ID	Nome	Valor	Nota
1	Julia	88	9
2	Carol	68	7
3	Maria	99	10
4	Andreia	78	8
5	Jaqueline	63	7
6	Marcela	81	9

Assim, os seguintes alunos obtiveram notas 8, 9 ou 10:
Maria (10)
Marcela (9)
Julia (9)
Andreia (8)

5. Uma aplicação possui quatro classes: **A**, **B**, **C** e **D**. As classes **B** e **C** são subclasses de **A**. Nas classes **A**, **B** e **C** existem atributos diferentes e um método chamado **verificarSaldo()** com a mesma assinatura mas que executa operações

interior do método **main()** da classe **D** foram digitadas as seguintes instruções:

```
A* obj = new B();
double v = obj->verificarSaldo();
A* obj1 = new C();
double v1 = obj1->verificarSaldo();
```

Quando essas linhas foram executadas, a variável **v** recebeu o valor **100.00** e a variável **v1** recebeu o valor **125.00**. Note que tanto **obj** como **obj1** são objetos do tipo **A**, porém, além de possuírem atributos diferentes, ao chamar o método **verificarSaldo()** por meio desses objetos, o retorno contido nas variáveis **v** e **v1** foi diferente. Isso mostra um exemplo de:

- Encapsulamento
- generalização
- sobrecarga de métodos
- polimorfismo
- herança múltipla

6. Faça um programa que calcule o tamanho de um string informado pelo usuário (**não usar nenhuma função para isso**, tal como ***strlen()*** ou ***lenght()***);

7. Informações sobre animais de estimação são mantidos em duas tabelas separadas:

```
TABLE dogs
id INTEGER NOT NULL PRIMARY KEY,
name VARCHAR(50) NOT NULL
```

0 de 7 respondidas

```
id INTEGER NOT NULL PRIMARY KEY,
name VARCHAR(50) NOT NULL
```

Escreva uma consulta SQL que selecione o nome de todos os animais de estimação de maneira distinta.

CONCLUÍDO