



Lista 1 - Everest

[← Voltar para Seleção de Exercícios e Provas](#)

Enunciado



temperaturas iguais de acordo com a mensagem: 'Sim, existem ZZ dias com temperatura média repetida.' (onde ZZ é a quantidade de registros repetidos) caso aquela lista de temperaturas possua mais de um dia com temperatura repetida, ou 'Não, não existem dias com temperatura média repetida.', caso não haja.

Solução

```
def elementos_repetidos(lista):  
    repetidos = list()  
    for i in lista:  
        if lista.count(i) > 1:  
            repetidos.append(i)  
    count = len(repetidos)  
    if(count > 0):  
        return f"Sim, existem  
    else:  
        return "Não, não exist
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 11 casos de teste

Enunciado



entrada, onde existe um único número distinto nesta lista e os demais são repetidos (uma ou mais vezes), crie uma função para encontrar o número que é único e retorná-lo.

Ex:

Entrada: [18, 19, 20

Saída: 21

Ex:

Solução

```
def numero_unico(lista):  
    for i in lista:  
        if lista.count(i) == 1:  
            return i
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 10 casos de teste

Enunciado



Solução

```
def soletrando_str(palavra):  
    return list(palavra)
```

Ex:

Entrada: amor

Saída: ["a", "m", "o"]

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 9 casos de teste

Enunciado



Joãozinho, por um concurso é de soletração de trás para frente. Dessa forma Joazinho vai precisar se dedicar e estudar muito para poder ir bem no concurso. Para ajudá-lo, crie uma função que recebe uma string, inverte-a e depois separa os caracteres em uma lista que deve ser o retorno da função.

Ex:

—

Solução

```
def soletrando_invertido_str(p):  
    return list(palavra[::-1])
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 8 casos de teste

Enunciado



Seu desafio é identificar as letras não repetidas de uma palavra. Para ajudá-lo no treino, crie uma função que recebe uma palavra e retorna o índice do primeiro caractere não repetido desta palavra. Caso não exista caractere único na palavra em questão, retornar -1.

Ex:

Entrada: `amor`

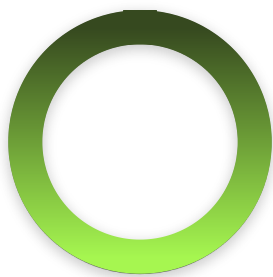
Solução

```
def primeiro_caractere_unico(palavra):  
    indice_unico = -1  
    for letra in palavra:  
        if palavra.count(letra) == 1:  
            indice_unico = palavra.index(letra)  
            break  
    return indice_unico
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 11 casos de teste

Enunciado



Solução

Coleções, Laços e Condicionais

Dentro dos conjuntos numéricos existem vários subconjuntos, dentre eles os subconjuntos dos números pares e números ímpares. Os números pares são todos os números múltiplos de 2, enquanto os números ímpares são o números não pares, logo, são os números que não são

```
s' if pares>1 else 'par'}, {impa
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 9 casos de teste

Enunciado



Solução

No dia a dia é comum trabalharmos com conjuntos de dados e em casos específicos pode ser necessário o uso de algum padrão de

```
def ordena(lista):  
    n = len(lista)  
    for i in range(n):  
        for j in range(n-i-1):  
            if lista[j] > list:
```

entender os dados, como ordem alfabética, por exemplo.

Crie uma função que recebe uma lista de dados (inteiros), faz a

```
lista[j], lista
return lista
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 9 casos de teste

Enunciado



Para ajudar aos alunos crie uma função que recebe uma lista de palavras e retorna uma lista com as palavras que são substring de qualquer outra string existente na lista. No retorno, as substrings devem ser ordenadas de acordo com a disposição na lista de entrada, aparecendo uma única vez. Caso não existam substrings, a função deve

Solução

```
def substring_str(lista):
    substrings = list()
    for i in lista:
        for j in lista:
            if (i != j) and (i
                substrings.appe

    return substrings
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 9 casos de teste

Enunciado



recebe em uma lista um dict com o nome do aluno, a média do aluno (nota) e a quantidade de horas (faltas) que o aluno teve durante a disciplina. Com esses dados, a função deve calcular se o aluno foi aprovado ou reprovado. Para calcular a presença do aluno considerar que o curso teve um total de 60 horas. O percentual deve ser calculado considerando a seguinte

Solução

```
def avaliacao(lista):  
    nome = lista[0]['nome']  
    nota = lista[0]['nota']  
    faltas = lista[0]['faltas']  
  
    total_horas = 60  
    presenca = (total_horas-fa  
    if (nota >= 7 and presenca  
        return f"{nome} está a  
    return f"{nome} está repro
```

Resultados

Sucesso:

Resposta Correta



O aluno obteve **100%** de **acerto** para os 7 casos de teste

