

1a. Prova de Programação de Computadores

Instruções para realização da prova:

- 1- A prova é individual e sem consulta à material, colegas, internet, ou qualquer outro meio de consulta
- 2- A prova poderá ser feita com auxílio do Pycharm
- 3- O aluno deverá copiar o código do Pycharm e colar na prova (formato .doc)
- 4- O aluno deverá devolver a prova antes do término do prazo
- 5- A prova tem duração de 2 horas
- 6- Se for detectado plágio ou qualquer outro tipo de desonestidade quanto à realização da prova, a mesma será zerada.

1a. Questão: Após o desastre que foi a prova de um professor, para não reprovar uma grande quantidade de alunos, ele decidiu normalizar as notas, alterando a maior nota para 10, e usando regra de três para a nota dos outros alunos. Por exemplo, se as notas forem 0, 3.4, 4.9 e 5.8, as notas normalizadas serão 0, 5.86, 8.45 e 10. Entretanto, ele não quer fazer a normalização manualmente. Faça um programa em Python para ajudar o trabalho desse professor. O programa deve receber a nota de 5 alunos e exibir as notas normalizadas na mesma ordem em que foram recebidas.

Sugestão: Seu programa deverá inicialmente encontrar o maior número. Este corresponderá ao valor 10. Em seguida, seu programa deverá efetuar a regra de 3 para obter as notas normalizadas.

Obs.: Você deverá resolver essa questão sem utilizar comandos de repetição

def normalizar_notas(notas):

```
    notas_normalizadas = []
```

```
    nota_max = max(notas)
    nota_max_normalizada = 10
```

```
    nota_1 = round(((notas[0]*nota_max_normalizada)/nota_max), 2)
    notas_normalizadas.append(nota_1)
```

```
    nota_2 = round(((notas[1]*nota_max_normalizada)/nota_max), 2)
    notas_normalizadas.append(nota_2)
```

```
    nota_3 = round(((notas[2]*nota_max_normalizada)/nota_max), 2)
    notas_normalizadas.append(nota_3)
```

```

nota_4 = round(((notas[3]*nota_max_normalizada)/nota_max), 2)
notas_normalizadas.append(nota_4)

nota_5 = round(((notas[4]*nota_max_normalizada)/nota_max), 2)
notas_normalizadas.append(nota_5)

return notas_normalizadas

```

NAO FAZ NADA

2a. Questão: Escreva um programa em Python que, dado um número de segundos, converta para dias, horas, minutos e segundos. Por exemplo, 7322 segundos correspondem a 0 dias, 2 horas, 2 minutos e 2 segundos.

def converter_segundos(segundos):

```

    dias = segundos/86400

    if len(str(dias)) == 3:
        if '.0' in str(dias):
            horas = 0
            minutos = 0
            segundos_convertido = 0
        else:
            horas = dias*24
    else:
        horas = dias*24

    if len(str(horas)) == 3:
        if '.0' in str(horas):
            minutos = 0
            segundos_convertido = 0
        else:
            minutos = (horas - round(horas, 0))*60
    else:
        minutos = (horas - round(horas, 0))*60

    if len(str(minutos)) == 3:
        if '.0' in str(minutos):
            segundos_convertido = 0
        else:
            segundos_convertido = (minutos - round(minutos, 0))*60
    else:
        segundos_convertido = (minutos - round(minutos, 0))*60

```

```
print(f'{segundos} segundos equivalem a {int(round(dias, 0))} dias, {int(round(horas, 0))} horas, {int(round(minutos, 0))} minutos e {int(round(segundos_convertido, 0))} segundos')
```

nao retorna nada

3a. Questão: Faça um programa que solicite que o usuário digite sua senha. O programa deverá verificar a validade da senha fornecida pelo usuário. A senha válida é o número 3456. Devem ser impressas as seguintes mensagens:

ACESSO PERMITIDO: caso a senha seja válida

ACESSO NEGADO: caso a senha seja inválida

nao retorna nada.

PQ USOU FUNCAO?

def validar_senha():

```
    senha = int(input('Insira sua senha: '))
```

```
    if senha == 3456:
```

```
        print('ACESSO PERMITIDO')
```

```
    else:
```

```
        print('ACESSO NEGADO')
```

4a. Questão: Desenvolva um programa em Python que receba do usuário os nomes dos dois times de uma partida de futebol. Além disso, o placar do jogo de futebol desses dois times (o número de gols de cada time) e informe se o resultado foi um empate, ou qual time foi o vencedor.

NAO RETORNA NADA

def resultado_partida():

```
    time1 = str(input('Insira o primeiro time da partida: '))
```

```
    time2 = str(input('Insira o segundo time da partida: '))
```

```
    gols1 = int(input(f'Insira o número de gols do {time1.lower().capitalize()}: '))
```

```
    gols2 = int(input(f'Insira o número de gols do {time2.lower().capitalize()}: '))
```

```
    if gols1 > gols2:
```

```
        print(f'Vitória do {time1.lower().capitalize()} por {gols1}x{gols2} sobre o {time2.lower().capitalize()}')
```

```
    elif gols1 < gols2:
```

```
        print(f'Vitória do {time2.lower().capitalize()} por {gols2}x{gols1} sobre o {time1.lower().capitalize()}')
```

```
elif gols1 == gols2:  
    print(f'Empate entre {time1} e {time2}')
```