

04 09 20
000000

Nome: Pedro Reynaldo Tome Silva
Registro Acadêmico: 1460482011092

Resoluções e Respostas

① Ele disse "Que não vale a pena você aprender uma linguagem de programação não se ela mudar sua cabeça. O Python tem uma filosofia diferente, ela é uma linguagem orientada a objetos, é uma linguagem que você pode programar estruturado, é uma linguagem que você tem multi paradigmas mas todos os aspectos que programar em Python tem uma filosofia, por isso bonito é melhor do que feio, explícito é melhor do que implícito, simples é melhor do que complexo". Já o import this, ao executar imediatamente conta uma pequena história a beleza sobre o python, com todos os característicos para o aprendiz.

04 07 20

③

$X = 1$

`fim = int(input('fim: '))`

`while X <= fim`

`print`

`X = X + 1`

`fim: 6`

1

2

3

4

5

6

④ Python é uma linguagem dinâmica e é orientada a objetos, é fortemente tipada, com uso de determinados elementos para a construção, abstração, interface, portabilidade e correção, mesmo em Python a sobreposição é direta, em a necessidade de criar estruturas para a troca de caracteres e atribuições múltiplas, já a divisão de um sobreclass em C seria mais complexa, pois além de fazer a conta de x ou outros variáveis para o valor, há a execução do resultado.

⑤ A diferença é que os contadores incrementam algo constante (um valor constante) por exemplo, imprimir

04 07 20

uma sequência de números, já os acumulados, os valores que não pode repetir e muda de acordo com as funções que os ordena, além disso ela representa a maior parte na confecção do programa, pois com ela começa a surgir, vários caracteres e atribui os finais de cada linha a executar o código.

(10) neste primeiro item (abaixo)

```
def duplo(f):
    for i in range(10):
        if str(i) + str(i) in f:
            return True
    return False
```

Nesta primeira parte do programa (item em) o sistema lê o código e executa e no meio destes números há dígitos duplicados (por exemplo, dois dígitos de um telefone de seis algarismos), mas antes dele ter e separar daqueles que possuem os que não possuem. Todos os caracteres devem ser convertidos para texto com o uso da função split()

```
def somapar(f):
    soma = 0
    for d in f:
        soma = soma + int(d)
    return soma % 2 == 0
```

Terminando (item dois) agora este novo programa deverá

Ver se todos os números somados e' divisivel por dois (a soma dos algarismos) e o resto tambem devera ser por (tudo o numero por e' divisivel por dois) e lembrar usar de uma funcao pra calcular estes numeros, ter um esquema que foram aprouveis pra para o item tres.

16) Para ter se pimeiro tenho perguntar todas juntas com o objetivo principal de se todos todos iguais, caso contrario a diferenca sera percebida nos codigos seguintes contendo dois dos tres (uso tres vezes com todos) e se ficar um do contrario, alem de ocorrer erros, Tomara acaba ficando em certo o codigo da funcao, resumindo isso o valor de tres juntas e o uso mais raro de ocorrer nos serve pra detectar alguma diferenca ou nao

30)

```
def fixa - primeiro(s):
    return s[0] + s[1:].replace(s[0], 'x')
```

Neste codigo o objetivo principal e' trocar o caracter de uma palavra por outro, mas se o primeiro ocorrer a partir de algum caracter em diante, para evitar que o primeiro caracter fique fixo, com isso usa s[0] mais s[1:] e o replace e que da a funcao de troca da letra por outras congelando a primeira letra de cada palavra, mas com o return

04.07.20

32

```
def zf(n):
    n = str(n)[::-1]
    k = 0
    while n[k] != '0':
        k = k + 1
    return k
a = int(str(n)[::-1])
return len(str(n)) - len(str(a))
```

Neste exercício há duas formas de chegar a resposta, a primeira que vai até `return k` e onde inicia a variável e logo em seguida na linha de baixo é feito o comprimento do texto da `str` e o próximo deve cortar o que o enunciado pede (letra `k`) em baixo há a função `while` e pede o que deve fazer por fim antes de retornar deve cortar quanto número gerou antes e como retorna a outra forma é com `int` e o mesmo uso `len` para saber o número e depois vai a uma função de subtração todos o número `(n - a)` com `len` e `str` dentro de parênteses ao retornar o mesmo o denominador do código (o que se finira)

04 09 20

33

0

```

def x_ordes(words):
    x = []
    outros = []
    for w in words:
        if w.startswith('x'): # w[0] == 'x'
            x.append(w)
        else:
            outros.append(w)
    return sorted(x) + sorted(outros)

```

Neste programa o principal objetivo é separar os palavras que começam com a letra x dos outros e isso é feito com a criação de dois coletores (um para cada grupo) depois segue o programa normal com os caracteres append sendo adicionados em x ou em outros e por fim o return com a nova ordem das palavras, a função if w.startswith tem a mesma função que w[0] e qualquer uma das duas funciona para o resultado final do código.

(3)

Def. demanda (inf):
 retorna $E^* \text{ grau } f(x, c)$, grau m em m variáveis (inf) [1.]

Nesta caso deve-se usar os operadores de todos os graus
 (coeficientes e multiplicar) e depois a função aritmética
 é o que falta. Então, a primeira a ser usada é a multiplicação
 depois o m com todos os pontos a seguir, que o primeiro elemento
 é a variável x e a segunda a variável y , para obter o valor
 e fazer a multiplicação do número como x com o segundo
 e depois com o mesmo primeiro elemento que nos dá o produto
 x e a variável y pelo fato de nos fornecer a soma para a variável
 desejada.