

Acrescentei comentários, estude o que escrevi para a sub.
Um possível Gabarito P2 (podem existir outras soluções)

1.
 - Muitos não fizeram em duas fases, como solicitado na questão
 - Quando a lista é vazia, vai sangrar a tela pela divisão por zero
 - Podem ter números negativos, cuidado com começar o maior_valor_encontrado com zero

```
def maior_abaixo_media(lista):
    # a) Calcula a média e cria uma nova lista com elementos menores que a média
    if not lista: # Lida com lista vazia
        return None

    media = sum(lista) / len(lista)

    menores_que_media = []
    for numero in lista:
        if numero < media:
            menores_que_media.append(numero)

    # b) Retorna o maior valor dessa nova lista
    if not menores_que_media: # Se não houver valores menores que a média
        return None

    maior_valor_encontrado = menores_que_media[0]
    for valor in menores_que_media:
        if valor > maior_valor_encontrado:
            maior_valor_encontrado = valor

    return maior_valor_encontrado
```

2.
 - Era igual ao exercício da P1, só que mudei pares para ímpares
 - Veja os comentários da P1, vários alunos erraram o que já foi comentado antes

```
def zigue_zague(s):
    resultado = ''
    for i in range(len(s)):
        if i % 2 == 1:
            resultado += s[i].upper()
        else:
            resultado += s[i]
    return resultado
```

3.
 - A maioria não reparou que o nome da função é o mesmo que a variável dentro (soma)
 - Questão muito parecida com a que caiu na P1, veja os comentários lá: float(p), indentação return, print tem texto + número

a) Erros encontrados:

O return total está dentro do loop for. Isso faz com que a função retorne o valor após a primeira iteração, sem somar os demais elementos da lista. O return deve estar fora do loop.
A variável total não foi inicializada antes de ser usada no cálculo total = total + p.
Uma variável soma com o mesmo nome da função.
print não deve receber uma atribuição
Dentro do print estamos somando uma string com um número
Os elementos da lista preços são strings (devido ao .split()). Para somá-los, eles precisam ser convertidos para números (float, neste caso).

b) Código corrigido:

```
def soma(lista):
    total = 0.0
    for p in lista:
        total = total + float(p)
    return total

precos = "19.90 35.00 12.50 9.90".split()
print("Total: R$" + str(soma(precos)))
```

4.
 - n é um inteiro, então não faz sentido tentar indexar um item com n[], só se ele for transformado para string antes
 - necessário inverter de alguma forma

```
def pares_finais(n):
    n = str(n)
    n = n[::-1]
    cont = 0
    for x in n:
```

```

    if int(x) % 2 == 0:
        cont = cont + 1
    else:
        break
return cont

```

5.

-alguns alunos apenas fizeram $p2 = p2 * 2$ sem somar num acumulador, note que isso é a quantidade de grãos de uma única casa

```

total = 0
p2 = 1
for k in range(64):
    total = total + p2
    p2 = p2 * 2
print (total)
# 18446744073709551615

```

6.

-note que esse exercício não é o mesmo que anagrama

```

def mesmas_letras(s1, s2): return set(s1) == set(s2)

```

ou

```

def mesmas_letras(s1, s2):
    for x in s1:
        if x not in s2:
            return False
    for x in s2:
        if x not in s1:
            return False
    return True

```

-preciso fazer dois for, para dar False nos casos 'casa', 'casal' e 'casal', 'casa'

-uma string pode estar dentro da outra, mas não são as mesmas letras

7.

-vários alunos esqueceram da condição de empate de letras, que era um dos exemplos mostrados

-existe um caso particular onde não existem vogais, nesse caso não altera nada

```

def maldição(s):
    vogais = 'aeiou'
    letra = ''
    maior = 0
    for x in vogais:
        cont = s.count(x)
        if cont > maior:
            maior = cont
            letra = x
    return s.replace(letra, '#') if letra else s

```

8.

-Usamos $\text{len}(s) - 2$ para não acessar uma posição que não existe (invalid index)

-Cuidado que existem vogais maiúsculas, como mostrado no exemplo

```

def sanduíche_com_vogais(s):
    cont = 0
    for k in range(len(s) - 2):
        a = s[k].lower()
        b = s[k+2].lower()
        if a in 'aeiou' and b in 'aeiou':
            cont = cont + 1
    return cont

```

9.

-Caso você não monte de alguma forma os diferentes, terá o risco de somar várias vezes o número, e o enunciado diz "apenas uma vez na soma final"

-Observe que os números não precisam ser consecutivos

```

def soma_duplas(nums):
    diferentes = []
    for x in nums:
        if x not in diferentes:
            diferentes.append(x)
    #ou diferentes = set(lista)
    duplas = []
    for x in diferentes:
        if nums.count(x) == 2:
            duplas.append(x)
    #ou duplas = [x for x in diferentes if nums.count(x) == 2]
    return sum(duplas)

```

Solução em uma linha:

```
return sum([x for x in set(nums) if nums.count(x) == 2])
```

10.

```
['Falso lógico', 'abacate', 'Falso lógico', {42: 'resposta'}, 'Falso lógico', [42], 'Falso lógico', 42, True, True]
```

Eu sou Feliz!

4

2

0

Vou aprender mais!

11.

```
def espelho_frase(frase): return frase[::-1]
```

Comentário: aplicação direta do Zen do Python "simples é melhor que complexo". Era a questão mais simples da prova.

Alguns alunos fizeram frase.split() antes de aplicar [::-1], nesse caso as palavras seriam separadas e invertidas, o que não é o mesmo que inverter todas as letras da frase.