

P2 Alg 2025-2

Nome e turma: _____

1. Explique o que faz a função abaixo e descreva, em suas palavras, como o código funciona. Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `near_five` e retorne `True` se `n` estiver a uma distância de no máximo 1 de um múltiplo de cinco. Exemplos: `near_five(5) → True`, `near_five(4) → True`, `near_five(7) → False`.

```
def near_ten(n):  
    return str(n % 10) in '01289'
```

2. Escreva uma função chamada `sanduche_de_consoantes(s)` que conta quantas vezes ocorre um sanduíche de consoantes em uma string. Um sanduíche de consoantes é uma sequência de três letras em que a primeira e a terceira são consoantes, e a letra do meio pode ser qualquer uma.

```
test(sanduche_de_consoantes('banana'),2) test(sanduche_de_consoantes('Brasil'), 2)  
test(sanduche_de_consoantes('aeiou'), 0) test(sanduche_de_consoantes(''), 0)
```

3. Explique o que faz a função abaixo e descreva, em suas palavras, como o código funciona. Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `diferenca_na_lista` e verifique se existe um par de números distintos da lista cuja diferença seja exatamente `n`. O número `n` pode ser positivo, zero ou negativo. Exemplos: `diferenca_na_lista(1, [1, 2, 3, 4]) → True`, `diferenca_na_lista(-1, [1, 2, 3, 4]) → True`, `diferenca_na_lista(-3, [1, 2, 3, 4]) → True`, `diferenca_na_lista(-5, [1, 2, 3, 4]) → False`, `diferenca_na_lista(0, [2, 2, 2, 2]) → False`.

```
def soma_na_lista(n, lista):  
    return n in [x + y for x in lista for y in lista if x != y]
```

4. Explique o que faz a função abaixo e descreva, em suas palavras, como o código funciona. Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `fixa_ultimo` e retorne uma nova string onde todas as ocorrências do último caractere da string original sejam trocadas por `'*'`, exceto a última. Exemplo: `fixa_ultimo('banana') → 'b*n*na'`

```
def fixa_primeiro(s):  
    return s[0] + s[1:].replace(s[0], '*')
```

5. Explique o que faz a função abaixo e descreva, em suas palavras, como o código funciona. Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `soma_impares_finais` e retorne a soma dos dígitos ímpares consecutivos que aparecem no final de um número inteiro positivo. Ex: `soma_impares_finais(3519) → 18`, `soma_impares_finais(77770) → 0`, `soma_impares_finais(12345) → 5`

```
def zf(n):  
    n = str(n)[::-1]  
    cont = 0  
    for c in n:  
        if c == '0':
```

```

        cont += 1
    else:
        break
return cont

```

6. Explique o que faz a função `ordena_diferente` e descreva, em suas palavras, como o código usa a função `sorted` com a cláusula `key=` para definir o critério de ordenação. Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `ordena_por_tamanho` e ordene a lista de nomes pelo comprimento de cada nome, do menor para o maior.

```

ordena_diferente('Ana Mariana João Bea'.split()) → ['Ana', 'Mariana', 'Bea', 'João']
ordena_por_tamanho('Ana Mariana João Bea'.split()) → ['Ana', 'Bea', 'João', 'Mariana']

```

```

def ordena_diferente(nomes):
    def last(x):
        return x[-1]
    return sorted(nomes, key=last)

```

7. Dada uma palavra `p` sem espaços, retorne uma palavra sem as letras repetidas. def `remove(p)`. Não é necessário manter a ordem das letras. def `remove(s)`. Ex.: `remove('banana') -> 'nab'`

8. Dada uma frase, inverta cada palavra individualmente, mantendo a ordem das palavras. def `inverte(frase)`.

```

inverte('ana e mariana gostam e banana') -> 'ana e anairam matsog e ananab'
inverte('python é divertido') -> 'nohtyp é odotrevnid'

```

9. Dê os valores de `a` para as saídas: `'A B'`, `'A'`, `'B'`, `'C'` e `'Feliz Natal!'`. Se achar que não há valor possível, justifique.

```

a = ?
if a > 10 and a % 6 == 3:
    print ('A', end = ' ')
elif a > 10 and a < 20:
    print ('B', end = ' ')
else:
    print ('C', end = ' ')

```

10. a) Explique o que faz a função **comum** abaixo e descreva, em suas palavras o que faz o código. b) qual é a saída de `comum('azul', 'amarelo')`, `comum(range(5), [1,3,5])`, e `comum('azul', ['a','b'])`. c) Em seguida, modifique apenas o código apresentado (sem criar outro do zero) para que ele passe a se chamar `nao_comum` e retorne uma lista com os elementos de `seq1` que **não** aparecem em `seq2`. Exemplo.: `nao_comum([1, 2, 3], [2, 3, 4]) -> [1]`. def `nao_comum(seq1, seq2)`.

```

def comum(seq1, seq2):
    return [x for x in seq1 if x in seq2]

```