P2 Alg 2024-2

Nome e turma: _____

1. Qual é a saída o seguinte código:

```
if 3 == 3.0: print ('Eu', end = ' ')
if 123 % 356254 == 123: print ('Sou', end = ' ')
if 3 == '3': print ('Muito', end = ' ')
if '' in 'abacate': print ('Feliz.', end = ' ')
if [3, 4] in [1, 2, [3, 4]]: print ('Vou', end = ' ')
if True: print ('Aprender', end = ' ')
if [1, 2, 3] and 42: print ('Mais!', end = ' ')
if False: print ('Até o próximo ano!', end = ' ')
```

2. def felicidade(repetições), onde repetições é uma lista com 5 inteiros positivos, que mostram quantas vezes cada letra da palavra 'FELIZ', irá aparecer na posição. Use repetições de alguma forma. Dentro da função use a variável feliz = 'FELIZ'
   Exemplo: felicidade([2, 3, 1, 2, 1] #mostra F F E E E L I I Z
3. O código abaixo foi escrito para contar quantas palavras em uma lista terminam com a letra 'a'. No entanto, o programa contém quatro erros distintos. a) Encontre os quatro erros no código. b) Refaça o código sem os erros.

```
def terminam(palavras):
    contador = 0
    for palavra in palavras:
        if palavra[-1] = 'a':
            contador = cont + 1
        return cont
palavras = "mesa abajur cadeira estante cama".split()
print("Número de palavras que terminam com a letra 'a' é: " + terminam(palavras))
```

4. Qual são os valores de x para que sejam mostrada as saídas:
   a) **O O X O** b) **X O O O** c) **O O O O** d) **O X X O**

```
x = ?
for i in range(4):
    if i == x:
        print("X", end=" ")
    else:
        print("O", end=" ")
```

5. **Enade 2011.** No livro "O Homem que Calculava", de Malba Tahan, um personagem desejava ganhar os grãos de trigos que fossem distribuídos sobre um tabuleiro de xadrez do seguinte modo: um grão na primeira casa do tabuleiro, o dobro (2) na segunda, novamente o dobro (4) na terceira, outra vez o dobro (8) na quarta, e assim por diante, até a sexagésima quarta casa do tabuleiro. Faça um algoritmo que calcule a **quantidade total** de

grãos de trigo necessários para realizar esta distribuição. Não use o operador ** de exponenciação neste exercício.

6. def inverte(frase). Faça uma função que receba uma frase como entrada, e devolva as palavras da frase invertidas. Ex.: inverte('batatinha quando nasce') -> 'nasce quando batatinha'.
7. def dma(s). Existem datas em dois formatos: 'dd-mm-aaaa', 'dd/mm/aaaa'. Retorne dia, mês e ano numa lista. Se não houver '-' ou '/' retorna a string original.Exemplos: dma('08-11-2024') -> ['08', '11', '2024'], dma('08/11/2024') -> ['08', '11', '2024'], dma('abobrinha') -> 'abobrinha'
8. def anagrama(s1, s2). Duas palavras são anagramas quando possuem as mesmas letras em outra ordem. Não usar repetições (for e while). Exemplos: anagrama('alegria', 'alergia') -> True, anagrama('sim', 'siiimmmmm') -> False, anagrama('palmeiras', 'abacate') -> False
9. def pi(n). Faça um programa que calcule o valor aproximado de pi com n termos, segundo a fórmula: 4/1 - 4/3 + 4/5 -4/7 + 4/9 - 4/11 …
10. def contar_ocorrencias(dna, sub) onde dna é uma sequência de DNA e sub uma subsequência de DNA. A função deve retornar o número de ocorrências da subsequência sub dentro de dna, considerando também casos em que as subsequências se sobrepõem. Não é permitido o uso do método count() para esta tarefa. Exemplos: contar_ocorrencias("ATATAT", "ATA") # Deve retornar 2 contar_ocorrencias("AAGGTTAGGTTAGG", "AGGT") # Deve retornar 3
11. Bônus. Escreva uma função `danca_das_letras(palavra)` que receba uma palavra como entrada e exiba cada letra dessa palavra "dançando" de uma forma especial: para cada letra, imprima-a em uma nova linha, adicionando um número de espaços antes dela igual à sua posição na palavra (começando de 0), criando um efeito de deslocamento em escada. Use uma repetição para percorrer cada letra da palavra e o operador `*` para adicionar os espaços antes de cada letra com base na sua posição. Por exemplo, se a palavra for `"Python"`, a saída deve ser:

```
P
 y
  t
   h
    o
     n
```