

MAC0110 Introdução à Computação
BACHARELADO EM ESTATÍSTICA, MATEMÁTICA E MATEMÁTICA APLICADA
Segunda Prova – 24 de maio de 2018

Nome: _____

Assinatura: _____

Nº USP: _____

Questão	Valor	Nota
1	2,0	
2	2,0	
3	3,0	
4	3,0	
Total	10,0	

Instruções:

1. Não destaque as folhas deste caderno. A prova pode ser feita a lápis.
2. A prova consta de 4 questões. Verifique antes de começar a prova se o seu caderno está completo.
3. As questões podem ser resolvidas em qualquer página. Ao escrever uma solução (ou parte dela) em página diferente do enunciado, escreva QUESTÃO X em letras ENORMES junto da solução.
4. As soluções devem ser em Python. **Você pode usar apenas recursos de Python vistos em aula.**
5. Você pode definir funções auxiliares e usá-las à vontade.
6. Cuidado com a legibilidade e, principalmente, com a TABULAÇÃO.
7. As soluções não precisam verificar consistência de dados.
8. Não é permitido o uso de folhas avulsas para rascunho, a consulta a livros, apontamentos, colegas ou equipamentos eletrônicos.
9. Desligue o seu celular e qualquer equipamento que possa perturbar o andamento da prova;

DURAÇÃO DA PROVA: 100 minutos



Questão 1 (2 pontos)

Suponha que o Python tenha lido as seguintes funções:

```
def a(x, y):           def b(x, y):           def c(x, y):           def d(x):
    return x * y        print x + y                x = [5, 6]            x[0] = "X"
                        y[0] = 7                return x+y            return x
```

A seguir está uma transcrição de uma seção do Python Shell. Complete as lacunas com o tipo e o valor do resultado da expressão correspondente. Se ocorrer um erro, escreva **ERRO**.

In [1]:	a(3, [1])	tipo:	_____	valor:	_____
In [2]:	a(3, 2.0)	tipo:	_____	valor:	_____
In [3]:	a(3, "x")	tipo:	_____	valor:	_____
In [4]:	a(3, 2)	tipo:	_____	valor:	_____
In [5]:	b([1,2], [3,4])	tipo:	_____	valor:	_____
In [6]:	x = [1, 2]				
In [7]:	y = [3, 4]				
In [8]:	c(x, y)	tipo:	_____	valor:	_____
In [9]:	x	tipo:	_____	valor:	_____
In [10]:	y	tipo:	_____	valor:	_____
In [11]:	s = "MAC0110"				
In [12]:	len(s)	tipo:	_____	valor:	_____
In [13]:	s[-1]	tipo:	_____	valor:	_____
In [14]:	s[7]	tipo:	_____	valor:	_____
In [15]:	d(s)	tipo:	_____	valor:	_____
In [16]:	x = [1, 'x', 3.14, True]				
In [17]:	z = x[:]				
In [18]:	z[0] += 1				
In [19]:	x[0]	tipo:	_____	valor:	_____
In [20]:	z	tipo:	_____	valor:	_____

Questão 2 (2 pontos)

Para o trecho de código a seguir indique a saída (= resultado de `print()`).

```
def main():
    nusp = int(input("Digite o seu número USP: "))
    y = nusp % 5
    print("nusp =", nusp, "y = ", y)
    a = [10, 20, 30, 40, 50]
    b = f(a, y)
    print("a: ", a)
    print("y: ", y)
    print("b: ", b)

def f(x, y):
    n = len(x)
    x[y] = 0
    y = y % 2 + 1
    for i in range(y, n, 2):
        x[i] = -1
    return y

main()
```

Escreva abaixo a saída do programa:

Questão 3 (3 pontos)

Escreva uma função `junta()` que respeite a especificação a seguir.

```
def junta(lstA, lstB):
    ''' (list, list) -> list
    Recebe duas listas lstA e lstB com números.
    Os números de cada lista estão em ordem não-decrescente.
    A função retorna uma lista contendo todos os números em lstA e em lstB
    em ordem crescente e sem repetições.

    Exemplos:
    >>> junta([2, 5], [1, 2, 2, 3])
    [1,2,3,5]

    >>> junta([-4, -4, 10, 20], [-40, -4, 5, 15, 20])
    [-40, -4, 5, 10, 15, 20]

    >>> junta([], [110])
    [110]
    '''
```

Na sua função `junta()` você deve utilizar, **obrigatoriamente**, a função `pertence()` que respeita a especificação a seguir.

```
def pertence(x, lst):
    ''' (int, list) -> bool
    Recebe um inteiro x e uma lista lst.
    Retorna True caso x seja um elemento de lst, e retorna False em caso contrário.

    Exemplos:
    >>> pertence(2, [1,2,3])
    True
    >>> pertence(2, [])
    False
    '''
```

Você **NÃO** deve escrever a função `pertence()`, deve apenas utilizá-la na função `junta()`.

Dica: use dois índices distintos, um para a lista `lstA` e outro para a lista `lstB`. Incremente um dos índices quando um de seus elementos for inserido na lista a ser retornada.

Questão 4 (vale 3,0 pontos)

Escreva uma função `substitua()` que respeite a especificação a seguir.

```
def substitua(s, velha, nova):
    ''' (str, str) -> str
    Recebe três strings: s, velha, e nova.
    Retorna uma string t que é obtida percorrendo-se a string s e
    substituindo-se cada ocorrência da string velha pela string nova,
    a medida que velha é encontrada em s.

    Exemplos:
    >>> substitua("Catarina viu uma tarantula em cima da tartaruga", "a", "-|.")
    'C-|.t-|.rin-|. viu um-|. t-|.r-|.ntul-|. em cim-|. d-|. t-|.rt-|.rug-|.'

    >>> substitua("Catarina viu uma tarantula em cima da tartaruga", "tar", "")
    'Caina viu uma antula em cima da uga'

    >>> substitua("comer correr conter c", "er c", ".")
    'com.orr.ont.'

    >>> substitua("nanananan", "nan", "x")
    'xaxan'
    '''
```

