



MAC2166 - Introdução à Computação - 2023S1

Avaliação Sub

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

Utilize caneta azul ou preta e preencha completamente a quadrícula.  
Exemplo: ■. Não use ☒.

**Turma:** (somente um número; consulte a pessoa responsável se não souber)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

← Marque as quadrículas ao lado para formar o seu número USP e escreva seu nome completo em letra legível na linha pontilhada abaixo. **Se seu número possui menos que 8 dígitos complete com zeros à esquerda.**

Nome: \_\_\_\_\_

.....

Esta prova tem duração de 120 minutos. Não desmonte a prova.

**Q1 [2,0 pontos]** Simule o código abaixo e selecione as opções correspondentes às saídas (S1 a S5) impressa do programa a partir dos valores da variáveis n, divisor e codigo. Se marcar alguma opção errônea, você poderá ter uma penalização de nota na questão (se nada marcar, ficará com zero na linha/opção não selecionada).

```

inicio = 4
fim = 18
codigo = 0
n = inicio

while n <= fim:
    gerar = (n%2!=0)
    divisor = 3

    while (divisor*divisor <= n) and gerar:
        if n % divisor == 0:
            gerar = False
            divisor += 2

    if gerar:
        codigo = codigo*10 + n
        print("%d %d %d"%(n,divisor,codigo))

    n += 1

```

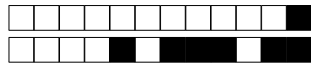
n	divisor	codigo

Abaixo estão as 5 opções para as possíveis saídas, S1 a S5, das variáveis que estão sendo impressas pelo programa. Em cada linha, selecione apenas uma opção.

Lembretes:

1. Se selecionar opção errônea para Si, então sua nota para a i-ésima impressão poderá ser negativa;
2. Se não marcar opção alguma para a saída Si, então receberá nota nula para a impressão i.

S1:	<input type="checkbox"/> 11 5 11	<input type="checkbox"/> 12 5 12	<input type="checkbox"/> 7 3 5	<input checked="" type="checkbox"/> 5 3 5	<input type="checkbox"/> 7 3 7
S2:	<input type="checkbox"/> 11 7 81	<input type="checkbox"/> 5 3 57	<input type="checkbox"/> 13 5 123	<input type="checkbox"/> 11 5 81	<input checked="" type="checkbox"/> 7 3 57
S3:	<input type="checkbox"/> 13 5 823	<input type="checkbox"/> 17 5 1247	<input type="checkbox"/> 11 7 1247	<input checked="" type="checkbox"/> 11 5 581	<input type="checkbox"/> 13 5 581
S4:	<input type="checkbox"/> 13 7 5823	<input type="checkbox"/> 17 5 8247	<input type="checkbox"/> 17 5 8247	<input checked="" type="checkbox"/> 13 5 5823	<input type="checkbox"/> 19 5 12489
S5:	<input checked="" type="checkbox"/> 17 5 58247	<input type="checkbox"/> 19 5 82489	<input type="checkbox"/> 23 5 124913	<input type="checkbox"/> 19 5 58247	<input type="checkbox"/> 23 7 124913



**Q2 [2,0 pontos]** Queremos fazer um programa que converte uma frase de entrada em outra, usando uma lista de pares de substituições de letras (de modo a converter algumas letras minúsculas para maiúsculas, por exemplo). Se queremos a conversão do caractere "i" para "I", especificamos em nossa lista o par ("i", "I"). Para tanto, definimos uma função `translitera(frase, pares)`, que recebe uma string contendo a frase de entrada e uma lista de pares (m, M) (ou seja, tuplas de tamanho 2) em que: m é um caractere minúsculo a ser substituído pelo maiúsculo M; a lista está ordenada crescentemente de acordo com a primeira componente m do par. A função `translitera` percorre a frase de entrada e converte cada caractere da frase que seja primeira componente de um par da lista pela segunda componente do par. Para tanto, `translitera` busca cada caractere da frase na lista ordenada de pares a fim de encontrar e substituir por seu substituto. Caso não seja encontrado na lista de pares, o caractere da frase não muda.

```
def translitera(frase, pares):
    nova_frase = ""
    for letra in L1:
        L2
        fim, inicio = L3, 0
        while fim >= inicio:
            L4
            minusc, maiusc = pares[meio]
            cod_minusc, cod_maiusc = ord(minusc), ord(maiusc)
            if cod_letra == cod_minusc:
                cod_letra = cod_maiusc
            L5
            elif L6:
                fim = meio - 1
            else:
                L7
            nova_frase += L8
    return nova_frase
```

**Execução:** Segue o exemplo de uma execução do programa onde o que é digitado como entrada para o programa está sublinhado:

Digite a frase: Goiabada de Araraquara  
 Transliterada: GOIAbAdA dE ArArAqUArA

```
vogais = [("a","A"),("e","E"),("i","I"),("o","O"),("u","U")]
frase = input("Digite a frase: ")
nova_frase = translitera(frase, vogais)
print("Transliterada: ", nova_frase)
```

Preencha as lacunas no código acima (L1 até L10), de forma a obter um programa em Python que realize a conversão de uma frase lida. OBS: Para cada lacuna, assinale no máximo uma resposta.

L1:	<input type="checkbox"/> nova_frase	<input checked="" type="checkbox"/> frase	<input type="checkbox"/> pares
L2:	<input checked="" type="checkbox"/> cod_letra = ord( letra )	<input type="checkbox"/> cod_letra = letra	<input type="checkbox"/> cod_letra = chr( letra )
L3:	<input type="checkbox"/> len( pares )	<input type="checkbox"/> len( frase )	<input type="checkbox"/> len( frase + pares )
L4:	<input checked="" type="checkbox"/> meio = ( inicio + fim ) // 2	<input type="checkbox"/> meio = inicio + 1	<input type="checkbox"/> meio = 1/2
L5:	<input checked="" type="checkbox"/> fim , inicio = meio-1 , meio	<input type="checkbox"/> fim = inicio	<input type="checkbox"/> inicio = fim
L6:	<input type="checkbox"/> cod_maiusc < meio	<input type="checkbox"/> cod_letra < meio	<input type="checkbox"/> cod_minusc < meio
L7:	<input type="checkbox"/> inicio = meio	<input checked="" type="checkbox"/> inicio = meio + 1	<input type="checkbox"/> inicio = ( inicio + fim + 1 ) // 2
L8:	<input type="checkbox"/> minusc + maiusc	<input checked="" type="checkbox"/> chr( cod_letra )	<input type="checkbox"/> letra



**Q3 [3,0 pontos]** No EP2, dado natural  $n > 0$ , um tabuleiro válido  $n \times n$  é uma permutação dos inteiros de 0 a  $n^2 - 1$ . Ao listar as casas de um tabuleiro da esquerda para a direita e de cima para baixo, forma-se uma sequência de números que é uma permutação de  $0, 1, \dots, n^2 - 1$ . Em contraste ao EP2, nesta questão o tabuleiro é representado pela lista que armazena esta sequência. Por exemplo, os tabuleiros  $3 \times 3$  a seguir são representados pelas listas  $[0, 1, 2, 3, 4, 5, 6, 7, 8]$  e  $[3, 5, 4, 1, 0, 2, 8, 6, 7]$ , respectivamente.

0	1	2
3	4	5
6	7	8

 e 

3	5	4
1	0	2
8	6	7

Assim, especificamos que a função `tabuleiro_valido( tab, n )` devolve:

- True, se a lista `tab` de tamanho `n` por `n` for uma permutação da lista  $[0, 1, 2, \dots, n \cdot n - 1]$ ;
- e False, caso contrário.

Dentre as opções de trecho a seguir, selecione cada uma que, quando substituída na [LACUNA] no esqueleto abaixo, forma o corpo de uma implementação correta da função `tabuleiro_valido`. Pode haver **mais de uma opção de trecho correta**; você deve selecionar todas as corretas. Haverá um **desconto na nota** a cada item **errôneo** marcado.

```
def tabuleiro_valido( tab, n )  
    [LACUNA]
```

```
t = list(tab) # copia  
t.sort() # ordena t  
k = 0  
for i in range(n*n):  
    if t[i] != i: k += 1  
return (k == 0)
```

```
t = list(tab) # copia  
for j in range(n*n):  
    k = j  
    for i in range(j+1, n*n):  
        if t[k] > t[i]: k = i  
        t[j] = t[k]  
        t[k] = t[j]  
for i in range(n*n):  
    if t[i] != i:  
        return False  
return True
```

```
t = list(tab) # copia  
for j in range(n*n):  
    k = j  
    for i in range(j+1, n*n):  
        if t[k] > t[i]: k = i  
        t[k] = t[j]  
        t[j] = t[k]  
for i in range(n*n):  
    if t[i] != i:  
        return False  
return True
```

```
t = list(tab) # copia  
t.sort() # ordena t  
v = len(tab) == n*n  
for i in range(n*n):  
    v = v and (t[i]==i)  
return v
```

```
t = list(tab) # copia  
for j in range(n*n):  
    k = j  
    for i in range(j+1, n*n):  
        if t[k] > t[i]: k = i  
        t[j], t[k] = t[k], t[j]  
for i in range(n*n):  
    if t[i] != i:  
        return False  
return True
```

```
val = True  
for i in range(n*n):  
    k = 0  
    for j in range(n*n):  
        if tab[j] == i:  
            k += 1  
    val = val and (k==1)  
return val
```

```
val = True  
for j in range(n*n):  
    k = 0  
    for i in range(n*n):  
        if tab[j] == i:  
            k += 1  
    val = val and (k==1)  
return val
```

```
t = list(tab) # copia  
for j in range(n*n):  
    k = j  
    for i in range(j+1, n*n):  
        if t[k] > t[i]: k = i  
        t[j], t[k] = t[k], t[j]  
for i in range(n*n):  
    if t[i] != i:  
        return False  
return True
```

Rascunho



**Q4 [3,0 pontos]** Escreva um programa em Python, completando as lacunas, que calcula a moda dos valores em uma matriz. A *moda* é o valor que ocorre com maior frequência em uma sequência de números. Assuma que os valores da matriz são sempre menores que a multiplicação de m por n, onde m e n são o número de linhas e colunas da matriz, respectivamente. A matriz deve ser passada como parâmetro e todas as linhas são compostas apenas por números inteiros. Por exemplo, para a matriz  $M = [[7, 5, 3], [7, 3, 3], [7, 5, 3]]$ , a moda é 3; e para uma simplificação do código, quando mais de um valor teve a mesma frequência, ou seja, houver um “empate” na moda, o menor valor será mostrado, como para  $M = [[0, 1, 2], [3, 4, 5], [6, 7, 8]]$ , a moda é 0.

```
def moda_matriz(matriz):
    L1
    L2
    L3
    count = [0]*tamanho
    L4
    L5
    count[matriz[i][j]] += 1
    L6
    L7
    freq = count[index]
    L8
    max_freq = freq
    L9
L10
def main():
    M = [[7, 5, 3], [7, 3, 3], [7, 5, 3]]
    print("Moda:",moda_matriz(M))
main()
```

Para cada um dos itens a seguir, correspondendo às lacunas no código acima, assinale a única resposta que torna o programa acima correto. As endentações do código correspondem a blocos do Python. Cada lacuna do programa corresponde a uma questão abaixo. Não tente montar o programa testando as combinações possíveis, pois não dará tempo. Escreva primeiro seu programa e depois procure analisar as opções abaixo. Isso é melhor do que tentar adivinhar o programa combinatoriamente olhando todas as alternativas. Cada opção errada que for selecionada será descontada na nota desta questão.

L1:	<input type="checkbox"/> m = len(matriz[0])	<input type="checkbox"/> m = 0	<input checked="" type="checkbox"/> m = len(matriz)	<input type="checkbox"/> len(matriz[0][0])	<input type="checkbox"/> NDA
L2:	<input type="checkbox"/> len(matriz[0][0])	<input type="checkbox"/> n = 0	<input type="checkbox"/> len(matriz)	<input checked="" type="checkbox"/> n = len(matriz[0])	<input type="checkbox"/> NDA
L3:	<input type="checkbox"/> tamanho = len(matriz)	<input type="checkbox"/> tamanho = m + n	<input type="checkbox"/> tamanho = len(matriz[0])	<input type="checkbox"/> tamanho = 0	<input checked="" type="checkbox"/> tamanho = m * n
L4:	<input type="checkbox"/> for i in range(tamanho):	<input type="checkbox"/> for i in range(m-1):	<input type="checkbox"/> for i in range(m+1):	<input checked="" type="checkbox"/> for i in range(m):	<input type="checkbox"/> NDA
L5:	<input checked="" type="checkbox"/> for j in range(n):	<input type="checkbox"/> for j in range(tamanho):	<input type="checkbox"/> for j in range(n-1):	<input type="checkbox"/> for j in range(n+1):	<input type="checkbox"/> NDA
L6:	<input type="checkbox"/> max_freq = 1	<input checked="" type="checkbox"/> max_freq = 0	<input type="checkbox"/> max_freq = tamanho	<input type="checkbox"/> max_freq = tamanho-1	<input type="checkbox"/> NDA
L7:	<input checked="" type="checkbox"/> for index in range(tamanho):	<input type="checkbox"/> for index in range(1,tamanho):	<input type="checkbox"/> NDA	<input type="checkbox"/> for index in range(tamanho+1):	<input type="checkbox"/> for index in range(tamanho-1):
L8:	<input type="checkbox"/> if freq == max_freq:	<input type="checkbox"/> if freq == moda	<input type="checkbox"/> if freq < moda	<input type="checkbox"/> if freq < max_freq:	<input checked="" type="checkbox"/> if freq > max_freq:
L9:	<input type="checkbox"/> NDA	<input checked="" type="checkbox"/> moda = index	<input type="checkbox"/> moda = max_freq	<input type="checkbox"/> max_freq = index	<input type="checkbox"/> freq = index
L10:	<input checked="" type="checkbox"/> return moda	<input type="checkbox"/> return index	<input type="checkbox"/> NDA	<input type="checkbox"/> return freq	<input type="checkbox"/> return max_freq: