



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"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10	<input type="checkbox"/> 11	<input type="checkbox"/> 12	<input type="checkbox"/> 20
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

← 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 [1 ponto] Simule o código abaixo e selecione as opções correspondentes à saída impressa do programa.

```
def f(L,e,d):
    c = (e+d)//2
    print(L[c])
    g(L,c+1,d)

def g(L,e,d):
    c = (e+d)//2
    for i in range(c, d):
        print(L[i])

def main():
    A = [14,42,35,87,65,23]
    f(A,0,5)
    g(A,1,4)

main()
```

Rascunho

Selecione a saída do primeiro comando **print** executado:

23 35 42 65 87 14

Selecione a saída do segundo comando **print** executado:

42 23 35 87 14 65

Selecione a saída do terceiro comando **print** executado ou N/A (não aplicável) se não ocorre mais que 2 impressões:

35 23 42 14 N/A 65 87

Selecione a saída do quarto comando **print** executado ou N/A (não aplicável) se não ocorre mais que 3 impressões:

35 65 14 N/A 23 87 42



Q2 [3 pontos] Assinale (pintando as quadrículas) os códigos que calculam a soma dos quatro vizinhos (superior, inferior, esquerdo e direito) de um dado elemento na linha y e coluna x de uma matriz A . O valor do elemento central não deve ser incluído na soma. Assuma que $0 < x < \text{len}(A[0])-1$ e $0 < y < \text{len}(A)-1$. A cada opção errada que for selecionada, desconta-se nota do exercício.

- ```
soma = 0
D = [[0,-1],[0,1],[-1,0],[1,0]]
for i in range(len(A)):
 for j in range(len(A[0])):
 if [i-y,j-x] in D:
 soma += A[i][j]
```
- ```
soma = 0
for i in range(y-1,y+2):
    for j in range(x-1,x+2):
        soma += A[i][j]
```
- ```
soma = 0
dx = [0, 1, 0, -1]
dy = [-1, 0, 1, 0]
for k in range(len(dx)):
 j = x + dx[k]
 i = y + dy[k]
 soma += A[i][j]
```
- ```
soma = 0
dx = dy = [0, 1, 0, -1]
for k in range(-1,len(dx)-1):
    j = x + dx[k]
    i = y + dy[k+1]
    soma += A[i][j]
```
- ```
soma = -A[y][x]
for i in range(y-1,y+2):
 for j in range(x-1,x+2):
 if i == x or j == y:
 soma += A[i][j]
```
- ```
soma = 0
for dy in [-1, 0, 1]:
    for dx in [-1, 0, 1]:
        if dy+dx == 1 or dy+dx == -1:
            soma += A[y+dy][x+dx]
```
- ```
soma = 0
D = [[0,-1],[0,2],[-1,-1],[2,0]]
for L in D:
 x += L[0]
 y += L[1]
 soma += A[y][x]
```
- ```
soma = 0
dx = dy = [0, 1, 0, -1]
for k in range(len(dx)):
    j = x + dx[k]
    i = y + dy[k+1]
    soma += A[i][j]
```
- ```
soma = 0
D = [[0,-1],[0,1],[-1,0],[1,0]]
for k in range(len(D)):
 x += D[k][0]
 y += D[k][1]
 soma += A[y][x]
```
- ```
soma = 0
for dy in range(-1,2):
    for dx in range(-1,2):
        if dy*dx == 0 and dy != dx:
            soma += A[y+dy][x+dx]
```
- ```
soma = 0
for dy in range(-1,1):
 for dx in range(-1,1):
 if dy*dx == 0 and dy != dx:
 soma += A[y+dy][x+dx]
```
- ```
soma = 0
for dy in [-1, 0, 1]:
    for dx in [-1, 0, 1]:
        if dy+dx not in [-1, 1]:
            soma += A[y+dy][x+dx]
```
- ```
soma = 0
for i in range(y-1,y+2):
 for j in range(x-1,x+2):
 if i == x or j == y:
 soma += A[i][j]
```
- ```
soma = 0
D = [[0,-1],[0,1],[-1,0],[1,0]]
for L in D:
    x += L[0]
    y += L[1]
    soma += A[y][x]
```
- ```
soma = 0
for i in range(y-1,y+2):
 for j in range(x-1,x+2):
 if i*j == 0 and i != j:
 soma += A[i][j]
```
- ```
soma = A[y][x+1] + A[y][x-1]
soma += A[y-1][x] + A[y+1][x]
```



Q3 [3 pontos] Seja **A** uma matriz de dimensões $p \times q$ e **B** uma matriz de dimensões $r \times s$. Então, o produto de Kronecker $\mathbf{A} \otimes \mathbf{B}$ é uma matriz de dimensões $pr \times qs$ da forma:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1s} & \dots & \dots & a_{1q}b_{11} & a_{1q}b_{12} & \dots & a_{1q}b_{1s} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2s} & \dots & \dots & a_{1q}b_{21} & a_{1q}b_{22} & \dots & a_{1q}b_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{r1} & a_{11}b_{r2} & \dots & a_{11}b_{rs} & \dots & \dots & a_{1q}b_{r1} & a_{1q}b_{r2} & \dots & a_{1q}b_{rs} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{p1}b_{11} & a_{p1}b_{12} & \dots & a_{p1}b_{1s} & \dots & \dots & a_{pq}b_{11} & a_{pq}b_{12} & \dots & a_{pq}b_{1s} \\ a_{p1}b_{21} & a_{p1}b_{22} & \dots & a_{p1}b_{2s} & \dots & \dots & a_{pq}b_{21} & a_{pq}b_{22} & \dots & a_{pq}b_{2s} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{p1}b_{r1} & a_{p1}b_{r2} & \dots & a_{p1}b_{rs} & \dots & \dots & a_{pq}b_{r1} & a_{pq}b_{r2} & \dots & a_{pq}b_{rs} \end{pmatrix}$$

Exemplo:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} = \begin{pmatrix} 1 \times 0 & 1 \times 5 & 2 \times 0 & 2 \times 5 \\ 1 \times 6 & 1 \times 7 & 2 \times 6 & 2 \times 7 \\ 3 \times 0 & 3 \times 5 & 4 \times 0 & 4 \times 5 \\ 3 \times 6 & 3 \times 7 & 4 \times 6 & 4 \times 7 \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{pmatrix}$$

Faça uma função que recebe duas matrizes reais **A** e **B** e devolve o produto de Kronecker $\mathbf{A} \otimes \mathbf{B}$ na matriz **C**.

```
def Kronecker (A,B):
    p,q = len(A), len(A[0])
    r,s = len(B), len(B[0])
    C = []
    L1: _____
    linha = []
    L2: _____
    linha.append(0)
    C.append(linha)
    L3: _____
    L4: _____
    L5: _____
    L6: _____
    L7: _____
    L8: _____
    L9: _____
    L10: _____
    L11: _____
    return C
```



Para cada um dos 11 itens a seguir, correspondendo as lacunas no código acima, assinale a única resposta que torna o programa acima correto. A cada opção errada que for selecionada, desconta-se nota do exercício.

L1:	<input type="checkbox"/> for i in range(q+s):	<input checked="" type="checkbox"/> for i in range(p*r):	<input type="checkbox"/> for i in range(q*s):
	<input type="checkbox"/> for i in range(p*s):	<input type="checkbox"/> for i in range(p+r):	
L2:	<input type="checkbox"/> for j in range(p*s):	<input type="checkbox"/> for j in range(q+s):	<input type="checkbox"/> for j in range(p*r):
	<input type="checkbox"/> for j in range(p+r):	<input checked="" type="checkbox"/> for j in range(q*s):	
L3:	<input checked="" type="checkbox"/> for i in range(p):	<input type="checkbox"/> for i in range(p+q):	<input type="checkbox"/> for i in range(p*q):
	<input type="checkbox"/> for i in range(r):		<input type="checkbox"/> for i in range(q*s):
L4:	<input type="checkbox"/> for j in range(q*s):	<input type="checkbox"/> for j in range(p*q):	<input type="checkbox"/> for j in range(p+q):
	<input type="checkbox"/> for j in range(s):	<input checked="" type="checkbox"/> for j in range(q):	
L5:	<input type="checkbox"/> col = i*q	<input type="checkbox"/> lin = i*p	<input type="checkbox"/> lin = i*s
	<input checked="" type="checkbox"/> lin = i*r	<input type="checkbox"/> col = i*p	<input type="checkbox"/> lin = i*q
L6:	<input type="checkbox"/> for k in range(r*s):	<input type="checkbox"/> for k in range(p*q):	<input checked="" type="checkbox"/> for k in range(r):
	<input type="checkbox"/> for k in range(s):		<input type="checkbox"/> for k in range(p*s):
L7:	<input type="checkbox"/> col = j*q	<input type="checkbox"/> col = j*r	<input type="checkbox"/> col = j*p
	<input checked="" type="checkbox"/> col = j*s	<input type="checkbox"/> col = j+s	
L8:	<input type="checkbox"/> for l in range(p*q):	<input checked="" type="checkbox"/> for l in range(s):	<input type="checkbox"/> for l in range(r*s):
	<input type="checkbox"/> for l in range(p):	<input type="checkbox"/> for l in range(q):	
L9:	<input type="checkbox"/> C[lin][col] = A[k][l]*B[k][l]	<input type="checkbox"/> C[lin][col] = A[i][j]*B[i][j]	<input type="checkbox"/> C[lin][col] = A[i][l]*B[k][j]
	<input type="checkbox"/> C[lin][col] = A[k][l]*B[i][j]	<input checked="" type="checkbox"/> C[lin][col] = A[i][j]*B[k][l]	
L10:	<input type="checkbox"/> lin = l+1	<input type="checkbox"/> lin += 1	<input type="checkbox"/> lin = col+1
	<input checked="" type="checkbox"/> col += 1	<input type="checkbox"/> col = lin+1	
L11:	<input checked="" type="checkbox"/> lin += 1	<input type="checkbox"/> col = l+1	<input type="checkbox"/> col = lin+1
	<input type="checkbox"/> col += 1	<input type="checkbox"/> lin = col+1	



Q4 [3 pontos] Preencha as lacunas no código abaixo (L1 até L10), de forma a obter um programa que recebe um string contendo exclusivamente palavras separadas por espaços em branco e determina a palavra mais frequente e sua frequência relativa de ocorrência no texto.

Exemplo: Para "ovo surpresa do kinder ovo" temos como saída: palavra: ovo freq. rel.: 0.4

```

def separa (texto) :
    L = L1
    p = ""
    for c in texto :
        if c == ' ' :
            L.append(p)
            p = ""
        else :
            p += c
    L2
    return L
def maxfreq (texto):
    L = L3
    pf = []
    f = L4
    for p in L:
        L5
        pf.append(p)
        f.append(1)
    L6
        for i in range(len(pf)):
            if p == pf[i]:
                L7
    maxpf = ""
    maxf = 0
    for i in range(len(pf)):
        if f[i] > maxf:
            L8
            L9
    return L10

def main ():
    texto = input('Digite as palavras: ')
    p, f = maxfreq(texto)
    print ('palavra:', p, 'freq. rel.:', f)

main()

```



Para cada um dos 10 itens a seguir, correspondendo as lacunas no código acima, selecione a única resposta que torna o programa acima correto.

L1: texto " [texto] [] 0

L2: L = [L] L = [L, p] L += c L += 1 L.append(p)

L3: texto texto[i] [] [separa(texto)] separa(texto)

L4: [] " L 0 [0]

L5: if not p in pf: if p[i] == pf[i]: if p != f: while p in pf: if p != pf:

L6: if pf[i] == f[i]: else: elif p not in pf: if not f in pf: if f == 0:

L7: pf.append(f) f[i] = 2 pf[i] = f[i] f[i] += 1 f.append(f[i]+1)

L8: maxf = i maxf = len(pf[i]) maxf = 0 maxf = f[len(f)-1] maxf = f[i]

L9: maxpf = str(maxf) maxpf = separa(texto)[i] maxpf = "" maxpf = separa(pf[i])[1] maxpf = pf[i]

L10: maxf/len(pf), maxpf maxf, len(L) maxpf, maxf maxpf, maxf/len(L) pf, f