



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)

4	5	6	7	8	9	10	11	12	13	14	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 [2 pontos] Simule o código abaixo e selecione as opções correspondentes a saída impressa do programa.

```
def f(A, B):
    for i in range(len(B)):
        print( A[B[i]] )

def g(A, B, p):
    soma = 0
    for i in range(p, p+3):
        soma += A[i]*B[i-p]
    return soma

def main():
    L = [12,24,35,41,57,69,70]
    K = [3,2,1]
    C = [0, 3]
    L.append(83)
    print(L[len(L)-4])
    f(L, C)
    for elem in C:
        d = g(L, K, elem)
        d = d % 8
        print(L[d])
    C[0] = C[0] + 1
    a,b = C[0],C[0]+2
    print( L[a:b] )
main()
```

Rascunho

Selecione o primeiro número impresso:

<input type="checkbox"/> 83	<input type="checkbox"/> 24	<input type="checkbox"/> 69	<input type="checkbox"/> 35	<input type="checkbox"/> 70	<input type="checkbox"/> 57	<input type="checkbox"/> 41	<input type="checkbox"/> 12
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Selecione o segundo número impresso:

<input type="checkbox"/> 24	<input type="checkbox"/> 57	<input type="checkbox"/> 70	<input type="checkbox"/> 83	<input type="checkbox"/> 12	<input type="checkbox"/> 35	<input type="checkbox"/> 69	<input type="checkbox"/> 41
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Selecione o terceiro número impresso:

<input type="checkbox"/> 69	<input type="checkbox"/> 24	<input type="checkbox"/> 35	<input type="checkbox"/> 83	<input type="checkbox"/> 41	<input type="checkbox"/> 70	<input type="checkbox"/> 12	<input type="checkbox"/> 57
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Selecione o quarto número impresso:

<input type="checkbox"/> 35	<input type="checkbox"/> 24	<input type="checkbox"/> 83	<input type="checkbox"/> 41	<input type="checkbox"/> 57	<input type="checkbox"/> 12	<input type="checkbox"/> 70	<input type="checkbox"/> 69
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Selecione o quinto número impresso:

<input type="checkbox"/> 41	<input type="checkbox"/> 70	<input type="checkbox"/> 35	<input type="checkbox"/> 12	<input type="checkbox"/> 69	<input type="checkbox"/> 24	<input type="checkbox"/> 83	<input type="checkbox"/> 57
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Selecione o resultado da sexta impressão:

<input type="checkbox"/> [35,41]	<input type="checkbox"/> [41,57]	<input type="checkbox"/> [69,70]	<input type="checkbox"/> [24,35]	<input type="checkbox"/> [57,69]	<input type="checkbox"/> [12,24]	<input type="checkbox"/> [70,83]
----------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------



Q2 [2.5 pontos] Queremos ler a nota de alunos de diversas turmas para determinar qual turma é a mais homogênea, ou seja, qual turma tem as notas com o menor desvio padrão. Para tanto, temos uma função `le_notas_calculando_media()` que lê as notas de uma turma, terminada por valor negativo e devolve uma dupla, onde o primeiro elemento é lista com essas notas e o segundo elemento é a média da turma. Temos também a função `calcula_desvio_padrao(notas,media)` que recebe uma lista de notas `notas` e sua média `media` e computa o desvio padrão das notas da turma, por meio do cálculo $\sqrt{\sum \frac{(nota_i - media)^2}{len(notas)}}$. Se tivermos duas turmas, uma com notas `[0,10.5]`, cuja média é 5 e desvio padrão 4.082, e outra turma com notas `[5,5,6,6]`, cuja média é 5.5 e desvio padrão 0.5, a turma 2 será a mais homogênea.

```
def le_notas_calculando_media():
    notas = []
    soma = 0
    nota=float(input("Nota do aluno %d:"%(1)))
    while L1:
        L2
        soma += nota
        nota=float(input("Nota do aluno %d:"%(len(notas)+1)))
    return L3

def calcula_desvio_padrao(notas,media):
    s=0
    for L4:
        L5
    return L6

n=int(input("Entre com o número de turmas: "))
turma_h, menor_desvio = -1, 20
for L7:
    print("--- Turma %d --"%turma)
    xs, m = L8
    L9
    if desv < menor_desvio:
        L10
print("Turma %d é mais homogênea com desvio %f"%(turma_h,menor_desvio))
```

Execução: Considere a seguinte execução do programa completo:

```
Entre com o número de turmas: 2
-- Turma 1 --
Nota do aluno 1: 0
Nota do aluno 2: 10
Nota do aluno 3: 5
Nota do aluno 4: -1
-- Turma 2 --
Nota do aluno 1: 5
Nota do aluno 2: 5
Nota do aluno 3: 6
Nota do aluno 4: 6
Nota do aluno 5: -1
Turma 2 é mais homogênea com desvio
0.500000
```

Utilizando duas funções auxiliares, preencha as lacunas no código acima (L1 até L10), de forma a obter um programa em Python conforme descrito acima, com a execução ao lado.

L1:	<input type="checkbox"/> <code>nota in range(n)</code>	<input type="checkbox"/> <code>nota >= 0</code>	<input type="checkbox"/> <code>nota > 0</code>	<input type="checkbox"/> <code>notas < 0</code>	<input type="checkbox"/> <code>nota in notas</code>
L2:	<input type="checkbox"/> <code>notas.append(nota)</code>	<input type="checkbox"/> <code>nota.adiciona(notas)</code>	<input type="checkbox"/> <code>append(notas, nota)</code>	<input type="checkbox"/> <code>nota += [notas]</code>	<input type="checkbox"/> <code>notas + [nota]</code>
L3:	<input type="checkbox"/> <code>soma/len(notas)</code>	<input type="checkbox"/> <code>notas,soma/len(notas)</code>	<input type="checkbox"/> <code>soma/len,notas</code>	<input type="checkbox"/> <code>notas,soma</code>	<input type="checkbox"/> <code>notas</code>
L4:	<input type="checkbox"/> <code>nota in notas</code>	<input type="checkbox"/> <code>nota in range(n)</code>	<input type="checkbox"/> <code>valor in notas</code>	<input type="checkbox"/> <code>i in range(n)</code>	<input type="checkbox"/> <code>notas in nota</code>
L5:	<input type="checkbox"/> <code>s = (nota -media)**0.5</code>	<input type="checkbox"/> <code>s = (nota-media)/2</code>	<input type="checkbox"/> <code>s /= nota -media</code>	<input type="checkbox"/> <code>s += abs(nota -media)</code>	<input type="checkbox"/> <code>s+=(nota-media)**2</code>
L6:	<input type="checkbox"/> <code>s**2</code>	<input type="checkbox"/> <code>(s/len(notas))**2</code>	<input type="checkbox"/> <code>(len(notas)/s)**0.5</code>	<input type="checkbox"/> <code>(s/len(notas))**0.5</code>	<input type="checkbox"/> <code>s**0.5</code>
L7:	<input type="checkbox"/> <code>i in range(turma)</code>	<input type="checkbox"/> <code>i in range(1,n+1)</code>	<input type="checkbox"/> <code>i in range(n)</code>	<input type="checkbox"/> <code>turma in range(1,n+1)</code>	<input type="checkbox"/> <code>turma in range(n)</code>
L8:	<input type="checkbox"/> <code>le_notas_calculando(media)</code>	<input type="checkbox"/> <code>le_notas_calculando_media(turma)</code>	<input type="checkbox"/> <code>le_notas_calculando(n)</code>	<input type="checkbox"/> <code>le_notas_calculando_media(n)</code>	<input type="checkbox"/> <code>le_notas_calculando_media()</code>
L9:	<input type="checkbox"/> <code>desv=calcula_desvio_padrao(notas,media)</code>	<input type="checkbox"/> <code>desv=calcula_desvio(xs,m)</code>	<input type="checkbox"/> <code>desv=calcula_desvio_padrao()</code>	<input type="checkbox"/> <code>desv=calcula_desvio_padrao(xs,m)</code>	<input type="checkbox"/> <code>desv+=calcula_desvio_padrao(xs,m)</code>
L10:	<input type="checkbox"/> <code>turma_h,menor_desvio=turma,desv</code>	<input type="checkbox"/> <code>menor_desvio=desv</code>	<input type="checkbox"/> <code>turma_h=turma</code>	<input type="checkbox"/> <code>turma,desv=turma_h,menor_desvio</code>	<input type="checkbox"/> <code>turma=turma_h</code>



Q3 [2.5 pontos] O programa abaixo auxila no controle de ocupação das salas de um cinema. O cinema possui várias salas, e cada sala tem uma grade de assentos representada por uma matriz 2D. Cada assento pode estar ocupado (representado pelo valor 1) ou livre (representado pelo valor 0).

O programa implementado é capaz de armazenar a ocupação de cada sala de cinema como uma matriz de inteiros (1 ou 0). Cada sala é representada por uma matriz, e todas as salas são armazenadas em uma lista.

A função chamada `calcular_ocupacao_sala(matriz_sala)` recebe uma matriz que representa uma sala e retorna a porcentagem de assentos ocupados. A função chamada `calcular_ocupacoes(cinema)` recebe a lista com todas as salas e retorna uma lista com a porcentagem de ocupação de cada uma.

O sistema deve exibir na tela a ocupação de cada sala com duas casas decimais, conforme mostrado no exemplo de execução.

```
def calcular_ocupacao_sala(matriz_sala):
    total_assentos = 0
    ocupados = 0
    for linha in matriz_sala:
        for assento in linha:
            L1
            if assento == 1:
                L2
        L3
    return 0.0
L4
```

```
def calcular_ocupacoes(cinema):
    L5
    for sala in cinema:
        L6
        L7
    L8
```

```
def main():
    cinema = [
        [1, 0, 1, 0],
        [1, 1, 0, 0],
        [0, 0, 0, 0]
    ],
    [
        [1, 1, 1],
        [1, 1, 1],
        [1, 0, 0]
    ],
    [
        [0, 0, 0],
        [0, 1, 0],
        [0, 0, 0]
    ]
    ]
    ocupacoes = calcular_ocupacoes(cinema)
    for i in range(len(ocupacoes)):
        print("Sala %d: %.2f%%"%(i,ocupacoes[i]))
    main()
```

Exemplo de execução do programa completo:

```
Sala 0: 33.33%
Sala 1: 77.78%
Sala 2: 11.11%
```

Preencha as lacunas indicadas assinalando as respostas correspondentes abaixo (apenas uma para cada linha). Não tente montar o programa testando as combinações possíveis, pois não vai dar tempo; procure primeiro identificar as principais variáveis do programa, seus significados e a estrutura do algoritmo utilizado.

L1: ☐ `total_assentos += 1` ☐ `total_assentos = total_assentos + 2` ☐ `total_assentos -= 1`
☐ `total_assentos = 0`

L2: ☐ `ocupados = ocupados + 2` ☐ `ocupados = total_assentos` ☐ `ocupados -= 1` ☐ `ocupados += 1`

L3: ☐ `if total_assentos == 0:` ☐ `if total_assentos != 0:` ☐ `if total_assentos > 0:`
☐ `if total_assentos < 0`

L4: ☐ `return (ocupados - total_assentos) * 100` ☐ `return ocupados * 100`
☐ `return (ocupados / total_assentos) * 100` ☐ `return (total_assentos / ocupados) * 100`

L5: ☐ `ocupacoes = [0]` ☐ `ocupacoes = None` ☐ `ocupacoes = 0` ☐ `ocupacoes = []`

L6: ☐ `ocupacao = sala` ☐ `ocupacao = len(sala)` ☐ `ocupacao = 0`
☐ `ocupacao = calcular_ocupacao_sala(sala)`

L7: ☐ `ocupacoes = ocupacoes + [[ocupacao]]` ☐ `ocupacoes.append(len(sala))` ☐ `ocupacoes.append(0)`
☐ `ocupacoes.append(ocupacao)`

L8: ☐ `return 0` ☐ `return None` ☐ `return ocupacoes` ☐ `return sum(ocupacoes)`



Q4 [3 pontos] Preencha as lacunas no código abaixo (L1 até L12), de forma a obter um programa que, dada uma matriz $n \times m$ com as notas de n alunos, numerados de 0 a $n-1$, em m avaliações, numeradas de 0 a $m-1$, calcula e exibe os três melhores alunos com maiores médias. As médias das avaliações de cada aluno devem ser calculadas pela média ponderada para pesos fornecidos. A saída do programa deve ser idêntica à do exemplo.

As seguintes duas funções devem ser implementadas:

- **calcula_medio(N, P):** dadas uma lista $N = [n_0, \dots, n_{m-1}]$, com m notas, e uma lista $P = [p_0, \dots, p_{m-1}]$, com m pesos, calcula e devolve a média ponderada pela fórmula: $\text{média} = \frac{\sum_{i=0}^{m-1} (n_i \cdot p_i)}{\sum_{i=0}^{m-1} p_i}$.
- **indice_tres_maiores(X):** dada uma lista $X = [x_0, \dots, x_{n-1}]$, com $n \geq 3$, devolve uma lista $I = [i_0, i_1, i_2]$ com os índices dos três maiores valores em X , tal que $x_{i_0} \geq x_{i_1} \geq x_{i_2} \geq x_k$, para $k \notin \{i_0, i_1, i_2\}$.

```
def calcula_medio(N, P):
    sn, sp = 0, 0
    for i in L1:
        L2
        L3
    return L4

def indice_tres_maiores(X):
    L5
    for k in L6:
        imax = -1
        for i in L7:
            if L8:
                if L9:
                    L10
        if imax != -1:
            L11
    return I
```

```
def main():
    Notas = [[9,3,5], #Notas do aluno 0
              [8,0,1], #Notas do aluno 1
              [4,6,2], #Notas do aluno 2
              [1,2,5], #Notas do aluno 3
              [2,6,8]] #Notas do aluno 4

    P = [1,2,2]
    M = []
    for N in Notas:
        M.append(calcula_medio(N, P))
    I = indice_tres_maiores(M)
    print("Três melhores classificados: ")
    for k in range(len(I)):
        print("aluno de índice", L12)

main()
```

Exemplo de execução do programa completo:

```
Três melhores classificados:
aluno de índice 4 com média 6.0
aluno de índice 0 com média 5.0
aluno de índice 2 com média 4.0
```

Para cada um dos **12** itens a seguir, correspondendo as lacunas no código acima, assinale a única resposta correta.

L1:	<input type="checkbox"/> range(len(N)-1)	<input type="checkbox"/> range(N)	<input type="checkbox"/> range(len(N)-1,0,-1)	<input type="checkbox"/> range(len(N))	<input type="checkbox"/> range(len(N)-1)	<input type="checkbox"/> range(len(N),-1,-1)
L2:	<input type="checkbox"/> sn = sn+N[i]	<input type="checkbox"/> sn = N[i]*P[i]	<input type="checkbox"/> sn = (sn+N[i])*P[i]	<input type="checkbox"/> sn += N[i]*P[i]	<input type="checkbox"/> sn += (N[i]*P[i])/sp	<input type="checkbox"/> sn += n_i*p_i
L3:	<input type="checkbox"/> sp += P[0:i]	<input type="checkbox"/> sp = sum(P[0:m])	<input type="checkbox"/> sp += p_i in P	<input type="checkbox"/> sp = sum(P[0:i])	<input type="checkbox"/> sp = P[i]	<input type="checkbox"/> sp += P[i]
L4:	<input type="checkbox"/> (sp/sn)	<input type="checkbox"/> (sn*(sp/sn))	<input type="checkbox"/> (sp)	<input type="checkbox"/> ((sn*sp)/sp)	<input type="checkbox"/> (sn)	<input type="checkbox"/> (sn/sp)
L5:	<input type="checkbox"/> I = 0	<input type="checkbox"/> I = -1	<input type="checkbox"/> I = [0,0,0]	<input type="checkbox"/> I = [0]	<input type="checkbox"/> I = []	<input type="checkbox"/> I = [-1]
L6:	<input type="checkbox"/> range(len(I)+1)	<input type="checkbox"/> range(len(X))	<input type="checkbox"/> I	<input type="checkbox"/> range(3)	<input type="checkbox"/> range(len(I))	<input type="checkbox"/> X
L7:	<input type="checkbox"/> range(X)	<input type="checkbox"/> range(len(X))	<input type="checkbox"/> range(len(I))	<input type="checkbox"/> X	<input type="checkbox"/> range(k)	<input type="checkbox"/> I
L8:	<input type="checkbox"/> X[i] not in I	<input type="checkbox"/> i != I[0]	<input type="checkbox"/> X[i] in I	<input type="checkbox"/> i == I[0]	<input type="checkbox"/> i in I	<input type="checkbox"/> i not in I
L9:	<input type="checkbox"/> imax != -1 and X[i] > X[imax]	<input type="checkbox"/> imax == -1 and X[i] > X[imax]	<input type="checkbox"/> i != imax and X[i] < X[imax]	<input type="checkbox"/> i > imax and X[i] != X[imax]	<input type="checkbox"/> imax != -1 or X[i] > X[imax]	<input type="checkbox"/> imax == -1 or X[i] > X[imax]
L10:	<input type="checkbox"/> imax = X[i]	<input type="checkbox"/> imax = i-1	<input type="checkbox"/> imax = k	<input type="checkbox"/> imax = i+1	<input type="checkbox"/> imax = I[k]	<input type="checkbox"/> imax = i
L11:	<input type="checkbox"/> imax = I.pop()	<input type="checkbox"/> del I[imax]	<input type="checkbox"/> I = [I[0], imax]	<input type="checkbox"/> I = [imax]	<input type="checkbox"/> I.append(imax)	<input type="checkbox"/> I = I + imax
L12:	<input type="checkbox"/> k,"com média",M[k]	<input type="checkbox"/> k,"com média",M[I[k]]	<input type="checkbox"/> I[k]-1,"com média",M[I[k]-1]	<input type="checkbox"/> I[k],"com média",M[I[k]]	<input type="checkbox"/> I[k]+1,"com média",M[I[k]+1]	