



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"/> 13	<input type="checkbox"/> 14	<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 [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)):
        d = A[B[i]]
        print( 7*d )

def g(A, B, p):
    d = len(B) // 2
    k = 0
    soma = 0
    for i in range(p-d, p+d+1):
        soma += A[i]*B[k]
        k += 1
    return soma

def main():
    L = [2,1,3,5,7,4,6]
    K = [3,-2,1]
    C = [2, 4]
    L.append(8)
    print(L[len(L)-4])
    f(L, C)
    for elem in C:
        d = g(L, K, elem)
        print(7*d)

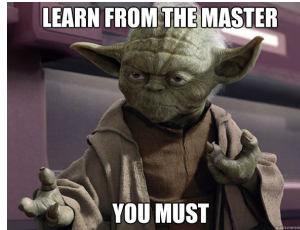
main()
```

Rascunho

- Selecione o primeiro número impresso:
- | | | | | | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <input type="checkbox"/> 8 | <input type="checkbox"/> 2 | <input type="checkbox"/> 6 | <input type="checkbox"/> 3 | <input type="checkbox"/> 7 | <input type="checkbox"/> 5 | <input type="checkbox"/> 4 | <input type="checkbox"/> 1 |
|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
- Selecione o segundo número impresso:
- | | | | | | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <input type="checkbox"/> 21 | <input type="checkbox"/> 42 | <input type="checkbox"/> 56 | <input type="checkbox"/> 63 | <input type="checkbox"/> 14 | <input type="checkbox"/> 28 | <input type="checkbox"/> 49 | <input type="checkbox"/> 35 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
- Selecione o terceiro número impresso:
- | | | | | | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <input type="checkbox"/> 49 | <input type="checkbox"/> 21 | <input type="checkbox"/> 28 | <input type="checkbox"/> 63 | <input type="checkbox"/> 35 | <input type="checkbox"/> 56 | <input type="checkbox"/> 14 | <input type="checkbox"/> 42 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
- Selecione o quarto número impresso:
- | | | | | | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <input type="checkbox"/> 28 | <input type="checkbox"/> 21 | <input type="checkbox"/> 63 | <input type="checkbox"/> 35 | <input type="checkbox"/> 42 | <input type="checkbox"/> 14 | <input type="checkbox"/> 56 | <input type="checkbox"/> 49 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
- Selecione o quinto número impresso:
- | | | | | | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <input type="checkbox"/> 35 | <input type="checkbox"/> 56 | <input type="checkbox"/> 28 | <input type="checkbox"/> 14 | <input type="checkbox"/> 49 | <input type="checkbox"/> 21 | <input type="checkbox"/> 63 | <input type="checkbox"/> 42 |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|



Q2 [2 pontos] Nesta questão visamos reproduzir o padrão de fala de alguns personagens de ficção. Para isso, implemente as seguintes funções em Python, assumindo o padrão ASCII e letras fornecidas maiúsculas:



(a) Cebolinha é um personagem de quadrinhos que quando falava, trocava o "R" pelo "L" (problema conhecido como dislalia). Faça uma função Cebolinha que gera uma versão modificada de um texto fornecido com todos "R" e "RR" trocados por "L", exceto no caso em que o "R" ocorre no final de uma palavra. Exemplo: "EU PODEREI COMPRAR RUAS E BAIROS INTEIROS" muda para "EU PODELEI COMPLAR LUAS E BAILOS INTEILOS".

(b) Faça uma função que altera o estilo do texto invertendo a ordem das palavras da frase imitando o mestre Jedi Yoda, movendo as duas primeiras palavras para o final da frase. Exemplos: "YOU WILL TALK LIKE YODA" muda para "TALK LIKE YODA YOU WILL". "YOU HAVE MUCH TO LEARN" vira "MUCH TO LEARN YOU HAVE".

```

def Cebolinha(T):
    S = ""
    for L1:
        if T[i] != 'R':
            L2
            L3
            if L4:
                c = 'L'
            else:
                c = 'R'
            L5
        L6
    return S

def Yoda(T):
    L = [] #Lista das palavras
    pal = ""
    for L7:
        if L8:
            L9
            pal = ""
        else:
            L10
    if pal != "":
        L11
    S = ""
    for L12:
        S += L[i]+' '
    S += L[0]+' '+L[1]
    return S

```

Preencha as lacunas nos códigos acima (L1 até L12).

L1:	<input type="checkbox"/> i in range(len(T)-1)	<input type="checkbox"/> i in range(len(T))	<input type="checkbox"/> i in range(1, len(T))			
L2:	<input type="checkbox"/> c = T[i-1]	<input type="checkbox"/> S += T[i]	<input type="checkbox"/> c = T[i]	<input type="checkbox"/> S[i] = T[i]	<input type="checkbox"/> S.append(T[i])	
L3:	<input type="checkbox"/> elif T[i-1] != 'R':	<input type="checkbox"/> elif T[i-1] == 'R':	<input type="checkbox"/> elif T[i+1] == 'R':	<input type="checkbox"/> else:	<input type="checkbox"/> elif T[i+1] != 'R':	
L4:	<input type="checkbox"/> T[i+1]>='A' and T[i+1]<='Z'	<input type="checkbox"/> T[i+1]<='A' and T[i+1]>='Z'	<input type="checkbox"/> T[i]>='A' and T[i]<='Z'	<input type="checkbox"/> T[i]>='A' or T[i]<='Z'	<input type="checkbox"/> T[i+1]>='A' or T[i+1]<='Z'	
L5:	<input type="checkbox"/> S.append(T[i])	<input type="checkbox"/> S.append(c)	<input type="checkbox"/> S += c	<input type="checkbox"/> S[i] = T[i]	<input type="checkbox"/> S += T[i]	<input type="checkbox"/> S[i] = c
L6:	<input type="checkbox"/> S += T[len(T)]	<input type="checkbox"/> S[i] = T[-1]	<input type="checkbox"/> S += T[len(T)-1]	<input type="checkbox"/> S += c	<input type="checkbox"/> S.append(T[-1])	
L7:	<input type="checkbox"/> c in range(len(T)-1)	<input type="checkbox"/> c in range(len(T))	<input type="checkbox"/> c in T	<input type="checkbox"/> c in T[1:]	<input type="checkbox"/> c in T[:len(T)-1]	
L8:	<input type="checkbox"/> pal != ""	<input type="checkbox"/> c == ' ' or pal != ""	<input type="checkbox"/> c != ' '	<input type="checkbox"/> c == ' ' and pal != ""	<input type="checkbox"/> c != ' ' and pal != ""	
L9:	<input type="checkbox"/> L += pal+c	<input type="checkbox"/> L.append(pal)	<input type="checkbox"/> L.append([pal])	<input type="checkbox"/> L = [pal] + L	<input type="checkbox"/> L.append(c)	
L10:	<input type="checkbox"/> pal.append(c)	<input type="checkbox"/> pal[len(pal)] = c	<input type="checkbox"/> pal = c + pal	<input type="checkbox"/> pal += [c]	<input type="checkbox"/> pal += c	
L11:	<input type="checkbox"/> L[len(L)] = pal	<input type="checkbox"/> L += pal	<input type="checkbox"/> L.append([pal])	<input type="checkbox"/> L.append(pal)	<input type="checkbox"/> L = [pal] + L	
L12:	<input type="checkbox"/> i in range(2, len(L))	<input type="checkbox"/> i in range(1, len(L)-1)	<input type="checkbox"/> i in range(len(L)-2)	<input type="checkbox"/> i in range(1, len(L))	<input type="checkbox"/> i in range(len(L)-1)	

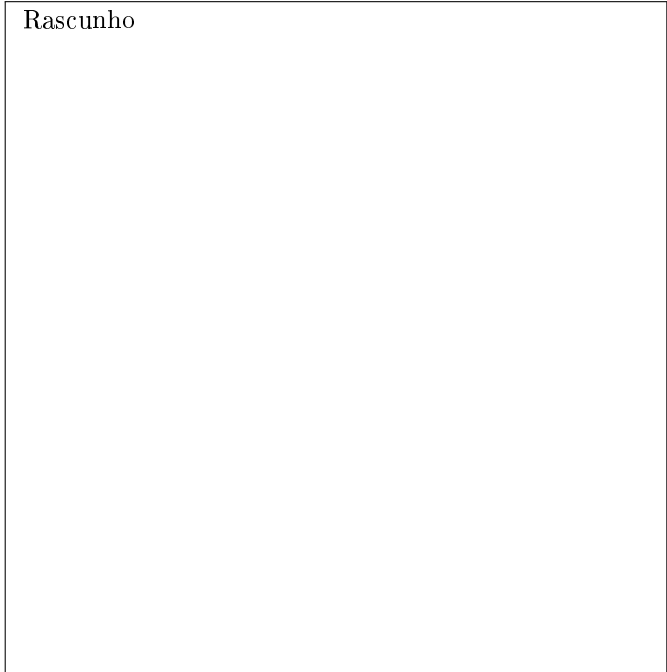


Q3 [3 pontos] Dada uma matriz T de caracteres, representando um tabuleiro de um jogo em que '0' indica o único jogador e 'X' indica os inimigos, faça a função DistanciaInimigos(T) em Python que retorna uma lista com dois elementos indicando o número de movimentos do jogador (seja para direita, esquerda, cima ou baixo) que são necessários para encontrar o inimigo mais próximo e o inimigo mais distante.

Exemplo: Para a entrada do programa main a saída impressa deve ser "dmin: 2, dmax: 4", pois o inimigo mais próximo está a 2 espaços do '0' (movendo um para direita e um para cima) e o inimigo mais distante está a 4 espaços do '0' (três para direita e um para cima). Dica: Use a função abs(a) que retorna o valor absoluto do número.

```
def DistanciaInimigos(T):
    inimigos = [] #Lista das posicoes inimigas
    for L1:
        for L2:
            if T[i][j] == '0':
                L3
                L4
                L5
        L6
    dmax = 0
    for L7:
        L8
        L9
        L10
        L11
    return [dmin, dmax]
```

```
def main():
    T = [[' ', 'X', ' ', 'X'],
         ['0', ' ', ' ', ' '],
         [' ', ' ', ' ', 'X']]
    D = DistanciaInimigos(T)
    print("dmin: %d, dmax: %d"%(D[0],D[1]))
main()
```



Preencha as lacunas no código acima (L1 até L11), assinalando a única resposta que torna o programa correto.

L1:	<input type="checkbox"/> i in range(len(T)-1)	<input type="checkbox"/> i in range(1, len(T))	<input type="checkbox"/> i in range(len(T[0])-1)
	<input type="checkbox"/> i in range(len(T))	<input type="checkbox"/> i in range(len(T[0]))	
L2:	<input type="checkbox"/> j in range(len(T)-1)	<input type="checkbox"/> j in range(1, len(T[0]))	<input type="checkbox"/> j in range(len(T[0])-1)
	<input type="checkbox"/> j in range(len(T))	<input type="checkbox"/> j in range(len(T[0]))	
L3:	<input type="checkbox"/> pi, pj = i, j+1	<input type="checkbox"/> pi, pj = i-1, j	<input type="checkbox"/> pi, pj = i, j-1
	<input type="checkbox"/> pi, pj = i, j		
L4:	<input type="checkbox"/> elif T[i][j] != T[pi][pj]:	<input type="checkbox"/> elif T[i][j] == 'X':	<input type="checkbox"/> elif T[pi][pj] == ' ':
	<input type="checkbox"/> elif T[i][j] == ' ':	<input type="checkbox"/> elif T[pi][pj] == 'X':	<input type="checkbox"/> else:
L5:	<input type="checkbox"/> inimigos.append([i+1,j])	<input type="checkbox"/> inimigos.append([i,j+1])	<input type="checkbox"/> inimigos.append([i,j-1])
	<input type="checkbox"/> inimigos.append([i,j])	<input type="checkbox"/> inimigos.append([i-1,j])	<input type="checkbox"/> inimigos = [i,j]
L6:	<input type="checkbox"/> dmin = len(T)+len(T[0])	<input type="checkbox"/> dmin = abs(pi-inimigos[0])+abs(pj-inimigos[1])	<input type="checkbox"/> dmin = 0
	<input type="checkbox"/> dmin = abs(pi - pj)	<input type="checkbox"/> dmin = len(T)	<input type="checkbox"/> dmin = len(T[0])
L7:	<input type="checkbox"/> k in range(len(inimigos)-1)	<input type="checkbox"/> k in range(1, len(inimigos))	<input type="checkbox"/> k in range(len(inimigos))
	<input type="checkbox"/> k in range(len(inimigos[0])-1)	<input type="checkbox"/> k in range(len(inimigos[0]))	
L8:	<input type="checkbox"/> di = abs(pi-inimigos[k])	<input type="checkbox"/> di = abs(pi-inimigos[k][0])	<input type="checkbox"/> di = abs(pi-inimigos[1][k])
	<input type="checkbox"/> di = abs(pi-inimigos[k][1])	<input type="checkbox"/> di = abs(pi-inimigos[0][k])	
L9:	<input type="checkbox"/> dj = abs(pj-inimigos[k][0])	<input type="checkbox"/> dj = abs(pj-inimigos[k][1])	<input type="checkbox"/> dj = abs(pj-inimigos[k])
	<input type="checkbox"/> dj = abs(pj-inimigos[1][k])	<input type="checkbox"/> dj = abs(pj-inimigos[0][k])	
L10:	<input type="checkbox"/> if di+dj > dmin: dmin = di+dj	<input type="checkbox"/> if di+dj < dmin: dmin = di+dj	<input type="checkbox"/> while di+dj < dmin: di = di+1
			<input type="checkbox"/> if abs(di-dj) < dmin: dmin = abs(di-dj)
L11:	<input type="checkbox"/> if di+dj < dmax: dmax = di+dj	<input type="checkbox"/> elif di+dj > dmax: dmax = di+dj	<input type="checkbox"/> if abs(di-dj) > dmax: dmax = abs(di-dj)
			<input type="checkbox"/> if di+dj > dmax: dmax = di+dj



Q4 [3 pontos] Considere os 4 trechos (T1 até T4) de código a seguir e depois selecione as afirmações verdadeiras sobre eles pintando as quadrículas. **Considerações:** **1.** As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas. **2.** A cada item errado que for selecionado, desconta-se nota do exercício.

T1:

```
def w(H,i,j):
    m,n = len(H),len(H[0])
    while 0<=i and i<m:
        while 0<=j and j<n:
            H[i][j] = 0
            j += 1
        i += 1
        j -= n
```

T2:

```
def p(B):
    m,n = len(B),len(B[0])
    i,j,k = 0,0,B[0][0]
    while i<m:
        if k<B[i][j]:
            k = B[i][j]
        i += 1
    while j<n:
        if k<B[i-1][j]:
            k = B[i-1][j]
        j += 1
    return k
```

T3:

```
def y(M,m,n):
    for i in range(m):
        for j in range(n):
            M[i][j] = M[j][i]
```

T4:

```
def f(A,i,j):
    y = A[i][j]
    r,s,n = i-1,j-1,0
    while n<8:
        y += A[r][s]
        if n%4<2:
            r += (-1)**(n//4)
        else:
            s += (-1)**(n//4)
        n += 1
    if r+s<i+j:
        return y
    else:
        return r+s
```

- T1
- w faz uma varredura por linhas começando em H[i][j]
 - w dá erro de execução se (i,j) não for uma posição válida da matriz H
 - se (i,j) for uma posição válida da matriz H, a instrução j -= n é equivalente a j = 0
 - w atribui 0 a todas as posições da matriz H
- T2
- p percorre a primeira coluna e a última linha de B
 - A primeira comparação k<B[i-1][j] produz False
 - p calcula o menor dentre os elementos da primeira coluna e da última linha de B
 - p calcula o maior dentre os elementos da primeira coluna e da penúltima linha de B
- T3
- Se M tem dimensão mxn com m≠n, ocorre um erro de execução
 - A função y transpõe a matriz M
 - Se M é uma matriz quadrada de ordem maior do que max(m,n), ocorre um erro de execução
 - Se M tem dimensão mxn e m=n, a função y reflete elementos abaixo da diagonal de M para cima
- T4
- Sendo A uma matriz mxn, f(A,i,j) não funciona nem na última linha nem na última coluna de A
 - f(A,i,j) calcula $\sum_{n=0}^7 -1^{n//4}$
 - f(A,i,j) sempre dá erro se A tem dimensão inferior a 8x8 para qualquer (i,j)
 - f(A,i,j) calcula $\sum_{\substack{i-1 \leq r \leq i+1 \\ j-1 \leq s \leq j+1}} A[r][s]$