

Centro de Educação Superior a Distância do  
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

# **Caderno de Exercícios**

## **Aula 5**

*(Vetor, Matriz, String (Cadeia de Caracteres) e Tuplas)*

Professores

Dante Corbucci Filho  
Leandro A. F. Fernandes

## Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta;
- Na saída, toda linha deve terminar com o caractere `'\n'` ;
- Utilize o URI Online Judge (<http://www.urionlinejudge.com.br>) e submeta sua solução para correção automática.

## Referências Autorais

Os exercícios apresentados nesta lista foram extraídos do URI Online Judge (<http://www.urionlinejudge.com.br>). Acesse a URL apresentada abaixo do título de cada problema para proceder com a correção automática de sua solução e, também, para consultar a autoria do enunciado.

## Problema A: Combinador

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1238>

Implemente um programa denominado combinador, que recebe duas strings e deve combiná-las, alternando as letras de cada string, começando com a primeira letra da primeira string, seguido pela primeira letra da segunda string, em seguida pela segunda letra da primeira string, e assim sucessivamente. As letras restantes da cadeia mais longa devem ser adicionadas ao fim da string resultante e retornada.

### Entrada

A entrada contém vários casos de teste. A primeira linha contém um inteiro N que indica a quantidade de casos de teste que vem a seguir. Cada caso de teste é composto por uma linha que contém duas cadeias de caracteres, cada cadeia de caracteres contém entre 1 e 50 caracteres inclusive.

### Saída

Combine as duas cadeias de caracteres da entrada como mostrado no exemplo abaixo e exiba a cadeia resultante.

### Exemplo

Entrada	Saída
2 Tpo oCder aa bb	TopCoder abab

## Problema B: Mensagem Oculta

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1272>

Textos podem conter mensagens ocultas. Neste problema a mensagem oculta em um texto é composto pelas primeiras letras de cada palavra do texto, na ordem em que aparecem.

É dado um texto composto apenas por letras minúsculas ou espaços. Pode haver mais de um espaço entre as palavras. O texto pode iniciar ou terminar em espaços, ou mesmo conter somente espaços.

### Entrada

A entrada contém vários casos de testes. A primeira linha de entrada contém um inteiro  $N$  que indica a quantidade de casos de teste que vem a seguir. Cada caso de teste consiste de uma única linha contendo de um a 50 caracteres, formado por letras minúsculas ('a'-'z') ou espaços (' '). Atenção para possíveis espaços no início ou no final do texto!

Nota: No exemplo de entrada os espaços foram substituídos por pequenos pontos ('.') para facilitar o entendimento dos exemplos.

### Saída

Para cada caso de teste imprima a mensagem oculta no texto de entrada.

### Exemplo

Entrada	Saída
4	
compete·online·design·event·rating	coder
·u····r·i··o····n·l··i····n··e···	urionline
.	
round··elimination·during··onsite··contest	redoc

## Problema C: Frequência de Letras

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1255>

Neste problema estamos interessados na frequência das letras em uma dada linha de texto.

Especificamente, deseja-se saber qual(is) a(s) letra(s) de maior frequência do texto, ignorando o “case sensitive”, ou seja maiúsculas ou minúsculas (sendo mais claro, “letras” referem-se precisamente às 26 letras do alfabeto).

### Entrada

A entrada contém vários casos de teste. A primeira linha contém um inteiro N que indica a quantidade de casos de teste. Cada caso de teste consiste de uma única linha de texto. A linha pode conter caracteres “não letras”, mas é garantido que tenha ao menos uma letra e que tenha no máximo 200 caracteres no total.

### Saída

Para cada caso de teste, imprima uma linha contendo a(s) letra(s) que mais ocorreu(ocorreram) no texto em minúsculas (se houver empate, imprima as letras em ordem alfabética).

### Exemplo

Entrada	Saída
3 Computers account for only 5% of the country's commercial electricity consumption. Input frequency letters	co inptu e

## Problema D: Quadrados

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2327>

Chama-se de quadrado mágico um arranjo, na forma de um quadrado, de  $N \times N$  números inteiros tal que todas as linhas, colunas e diagonais têm a mesma soma.

Por exemplo, o quadrado abaixo

2	7	6
9	5	1
4	3	8

é um quadrado mágico de soma 15, pois todas as linhas ( $2 + 7 + 6 = 15$ ,  $9 + 5 + 1 = 15$  e  $4 + 3 + 8 = 15$ ), colunas ( $2 + 9 + 4 = 15$ ,  $7 + 5 + 3 = 15$  e  $6 + 1 + 8 = 15$ ) e diagonais ( $2 + 5 + 8 = 15$  e  $6 + 5 + 4 = 15$ ) têm a mesma soma (15).

Escreva um programa que, dado um quadrado, determine se ele é mágico ou não e qual a soma dele (caso seja mágico).

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha da entrada de cada caso de teste contém um inteiro  $N$  ( $2 < N < 10$ ). As  $N$  linhas seguintes contêm  $N$  inteiros cada, separados por

exatamente um espaço em branco. Os inteiros dentro do quadrado são todos maiores que 0 (zero) e menores que 1.000.

### Saída

Seu programa deve imprimir, na saída padrão, uma única linha com um inteiro representando a soma do quadrado mágico ou  $-1$  caso o quadrado não seja mágico.

### Exemplo

Entrada	Saída
3 2 7 6 9 5 1 4 3 8	15
3 1 2 3 4 5 6 7 8 9	-1
4 16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1	34

## Problema E: Justificador

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1273>

Nós temos algumas palavras e queremos justificá-las à direita, ou seja, alinhar todas elas à direita. Crie um programa que, após ler várias palavras, reimprima estas palavras com suas linhas justificadas à direita.

### Entrada

A entrada contém diversos casos de testes. A primeira linha de cada caso de teste conterá um inteiro  $N$  ( $1 \leq N \leq 50$ ), que indicará o número de palavras que virão a seguir. Cada uma das  $N$  palavras contém no mínimo uma letra e no máximo 50 letras maiúsculas ('A'-'Z'). O fim da entrada é indicado por  $N = 0$ .

### Saída

Para cada caso de teste imprima as palavras inserindo tantos espaços quanto forem necessários à esquerda de cada palavra, para que elas apareçam todas alinhadas à direita e na mesma ordem da entrada. Deixe uma linha em branco entre os casos de teste. Não deixe espaços sobrando no final de cada linha nem imprima espaços desnecessários à esquerda, de modo que pelo menos uma das linhas impressa em cada texto inicie com uma letra.

### Exemplo

Entrada	Saída
3 BOB TOMMY JIM 4 JOHN JAKE ALAN BLUE 4 LONGEST A LONGER SHORT 0	BOB TOMMY JIM  JOHN JAKE ALAN BLUE  LONGEST A LONGER SHORT



## Problema F: Sudoku

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1383>

O jogo de Sudoku espalhou-se rapidamente por todo o mundo, tornando-se hoje o passatempo mais popular em todo o planeta. Muitas pessoas, entretanto, preenchem a matriz de forma incorreta, desrespeitando as restrições do jogo.

Sua tarefa neste problema é escrever um programa que verifica se uma matriz preenchida é ou não uma solução para o problema.

A matriz do jogo é uma matriz de inteiros 9 x 9. Para ser uma solução do problema, cada linha e coluna deve conter todos os números de 1 a 9. Além disso, se dividirmos a matriz em 9 regiões 3 x 3, cada uma destas regiões também deve conter os números de 1 a 9. O exemplo abaixo mostra uma matriz que é uma solução do problema.

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

### Entrada

São dadas várias instâncias. O primeiro dado é o número  $n > 0$  de matrizes na entrada. Nas linhas seguintes são dadas as  $n$  matrizes. Cada matriz é dada em 9 linhas, em que cada linha contém 9 números inteiros.

### Saída

Para cada instância seu programa deverá imprimir uma linha dizendo "*Instancia k*", onde  $k$  é o número da instância atual. Na segunda linha, seu programa deverá imprimir

"SIM" se a matriz for a solução de um problema de Sudoku, e "NAO" caso contrário. Imprima uma linha em branco após cada instância.

### Exemplo

Entrada	Saída
2	Instancia 1
1 3 2 5 7 9 4 6 8	SIM
4 9 8 2 6 1 3 7 5	
7 5 6 3 8 4 2 1 9	Instancia 2
6 4 3 1 5 8 7 9 2	NAO
5 2 1 7 9 3 8 4 6	
9 8 7 4 2 6 5 3 1	
2 1 4 9 3 5 6 8 7	
3 6 5 8 1 7 9 2 4	
8 7 9 6 4 2 1 5 3	
1 3 2 5 7 9 4 6 8	
4 9 8 2 6 1 3 7 5	
7 5 6 3 8 4 2 1 9	
6 4 3 1 5 8 7 9 2	
5 2 1 7 9 3 8 4 6	
9 8 7 4 2 6 5 3 1	
2 1 4 9 3 5 6 8 7	
3 6 5 8 1 7 9 2 4	
8 7 9 6 4 2 1 3 5	