

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 07**

*(Moda e Algoritmos de Ordenação)*

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: Ordenação por Tamanho

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1244>

Crie um programa para ordenar um conjunto de strings pelo seu tamanho. Seu programa deve receber um conjunto de strings e retornar este mesmo conjunto ordenado pelo tamanho das palavras, se o tamanho das strings for igual, deve-se manter a ordem original do conjunto.

### Entrada

A primeira linha da entrada possui um único inteiro N, que indica o número de casos de teste. Cada caso de teste poderá conter de 1 a 50 strings inclusive, e cada uma das strings poderá conter entre 1 e 50 caracteres inclusive. Os caracteres poderão ser espaços, letras, ou números.

### Saída

A saída deve conter o conjunto de strings da entrada ordenado pelo tamanho das strings. Um espaço em branco deve ser impresso entre duas palavras.

### Exemplo

Entrada	Saída
4 Top Coder comp Wedn at midnight one three five I love Cpp sj a sa df r e w f d s a v c x z sd fd	midnight Coder comp Wedn Top at three five one love Cpp I sj sa df sd fd a r e w f d s a v c x z

## Problema B: Plano de Dieta

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1248>

O doutor deu a você a sua dieta, na qual cada caractere corresponde a algum alimento que você deveria comer. Você também sabe o que você tem comido no café da manhã e no almoço, nos quais cada caractere corresponde a um tipo de alimento que você deveria ter comido aquele dia. Você decidiu que irá comer todo o restante de sua dieta durante o jantar, e você quer imprimi-la como uma String (ordenada em ordem alfabética). Se você trapaceou de algum modo (ou por comer muito de tipo de alimento, ou por comer algum alimento que não está no plano de dieta), você deveria imprimir a cadeia "CHEATER" (significa trapaceiro), sem as aspas.

### Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro  $N$  que indica a quantidade de casos de teste. Cada caso de teste é composto por três linhas, cada uma delas com uma string com até 26 caracteres de 'A'-'Z' ou vazia, representando respectivamente os alimentos da dieta, do café da manhã e do almoço.

### Saída

Para cada caso de teste imprima uma string que representa os alimentos que você deveria consumir no jantar, ou "CHEATER" caso você tenha trapaceado na sua dieta.

### Exemplo

Entrada	Saída
5 ABCD AB C ABEDCS  EDSMB MSD A  IWANTSODER SOW RAT	D ABCDES CHEATER  DEIN

## Problema C: Cifra de César

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1253>

Júlio César usava um sistema de criptografia, agora conhecido como Cifra de César, que trocava cada letra pelo equivalente em duas posições à Esquerda no alfabeto (por exemplo, 'C' vira 'A', 'T' vira 'R', etc.). Ao começo do alfabeto nós voltamos para o fim, isto é 'A' vira 'Y'. Nós podemos, é claro, tentar trocar as letras com quaisquer número de posições.

### Entrada

A entrada contém vários casos de teste. A primeira linha de entrada contém um inteiro **N** que indica a quantidade de casos de teste. Cada caso de teste é composto por duas linhas. A primeira linha contém uma string com até 50 caracteres maiúsculos ('A'-'Z'), que é a sentença após ela ter sido codificada através desta Cifra de César modificada. A segunda linha contém um número que varia de 0 a 25 e que representa quantas posições cada letra foi deslocada para a direita.

### Saída

Para cada caso de teste de entrada, imprima uma linha de saída com o texto decodificado (transformado novamente para o texto original) conforme as regras acima e o exemplo abaixo.

### Exemplo

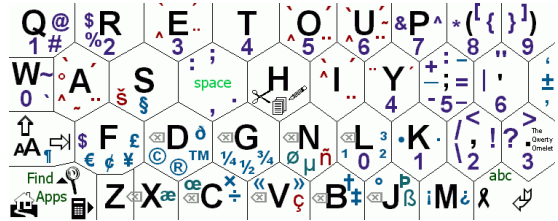
Entrada	Saída
6 VQREQFGT	TOPCODER QRSTUVWXYZABCDEFGHIJKLMNO
2 ABCDEFGHIJKLMNOPQRSTUVWXYZ	P TOPCODER
Z	AXCHMA
10 TOPCODER	CAMOBAP HELLOWORLD
0 ZWBGLZ	
25 DBNPCBQ	
1 LIPPSASVPH	
4	

## Problema D: 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 F: Pares e Ímpares

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1259>

Considerando a entrada de valores inteiros não negativos, ordene estes valores segundo o seguinte critério:

- Primeiro os Pares
- Depois os Ímpares

Sendo que deverão ser apresentados os pares em ordem crescente e depois os ímpares em ordem decrescente.

### Entrada

A primeira linha de entrada contém um único inteiro positivo  $N$  ( $1 < N < 10^5$ ). Este é o número de linhas de entrada que vem logo a seguir. As próximas  $N$  linhas conterão, cada uma delas, um valor inteiro não negativo.

### Saída

Apresente todos os valores lidos na entrada segundo a ordem apresentada acima. Cada número deve ser impresso em uma linha, conforme exemplo abaixo.

### Exemplo

Entrada	Saída
10	4
4	32
32	34
34	98
543	654
3456	3456
654	6789
567	567
87	543
6789	87
98	