

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 7

(Moda, Ordenação, Complexidade)

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: Organizador de Vagões

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1162>

Na estação de trem você ainda pode encontrar o último dos “organizadores de vagões”. Um Organizador de vagões um empregado cujo trabalho é apenas reordenar os vagões do trem, trocando-os de posição. Uma vez que os vagões são organizados em uma ordem considerada ótima, o condutor pode desconectar cada vagão e colocá-los na estação.

O título “organizador de vagões” é dado à pessoa que realiza esta tarefa, cuja estação fica perto de uma ponte. Ao invés da ponte poder subir ou descer, ela roda sobre um pilar que fica no centro do rio. Após rodar 90 graus, os barcos podem passar na esquerda ou direita dela. O Primeiro organizador de vagões descobriu que girando a ponte 180 graus com dois vagões em cima dela, é possível a troca de lugar entre os dois vagões. Obviamente a ponte pode operar no máximo com dois vagões sobre ela.

Agora que quase todos os organizadores de vagões já faleceram, a estação gostaria de automatizar esta operação. Parte do programa a ser desenvolvido é uma rotina que decide para um dado trem com um determinado número de vagões, o número de trocas entre trens adjacentes que são necessárias para que o trem fique ordenado. Sua tarefa é criar tal rotina.

Entrada

A entrada contém na primeira linha o número de caso de testes (N). Cada caso de teste consiste de duas linhas de entrada. A primeira linha de um caso de teste contém um inteiro L , determinando o tamanho do trem ($0 \leq L \leq 50$). A segunda linha de um caso de teste contém uma permutação dos números 1 até L , indicando a ordem corrente dos vagões. Os vagões devem ser ordenados de forma que o vagão 1 venha por primeiro, depois o 2, etc, com o vagão L vindo por último.

Saída

Para cada caso de teste imprima a sentença: 'Optimal train swapping takes S swaps.' onde S é um inteiro.

Exemplo

Entrada	Saída
3	Optimal train swapping takes 1 swaps.
3	Optimal train swapping takes 6 swaps.
1 3 2	Optimal train swapping takes 1 swaps.
4	
4 3 2 1	
2	
2 1	

Problema B: Primeiro Dicionário de Andy

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1215>

Andy de apenas 8 anos tem um sonho: ele deseja criar o seu próprio dicionário. Isto não é uma tarefa fácil para ele, pois conhece poucas palavras. Bem, em vez de pensar nas palavras que sabe, ele teve uma ideia brilhante. A partir do seu livro de histórias favorito, ele vai criar um dicionário com todas as palavras distintas que existem nele. Ordenando estas palavras em ordem alfabética, o trabalho estará feito. É claro, isso é uma tarefa que toma um certo tempo e portanto, a ajuda de um programador de computador como você é muito bem-vinda.

Você foi convidado a escrever um programa que liste todas as diferentes palavras que existem em um texto. Neste caso, uma palavra é definida como uma sequência de letras, maiúsculas ou minúsculas. Palavras com apenas uma letra também deverão ser consideradas. Portanto, seu programa deverá ser “*CaSe InSeNsItIvE*”. Por exemplo, palavras como “Apple”, “apple” ou “APPLE” deverão ser consideradas como a mesma palavra.

Entrada

A entrada contém no máximo 10000 linhas de texto, cada uma delas com no máximo 200 caracteres.

Saída

Você deve imprimir uma lista de diferentes palavras que aparecem no texto, uma palavra por linha. Todas as palavras devem ser impressas com letras minúsculas, em ordem alfabética. Deverá haver no máximo 5000 palavras distintas.

Exemplo

Entrada	Saída
Ex(*\$a#.mpl.e: Adventures in Disneyland Two blondes were going to Disneyland when they came to a fork in the road. The sign read: "Disneyland Left." So they went home.	a adventures blondes came disneyland e ex fork going home in left mpl read road sign so the they to two went were when

Problema C: 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	

Problema D: Fila do Recreio

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1548>

Na escola onde você estuda, a hora do recreio é a mais aguardada pela grande maioria dos alunos. Não só porque as vezes as aulas são cansativas, mas sim porque a merenda servida é muito boa, preparada por um chefe italiano muito caprichoso.

Quando bate o sinal para a hora do recreio, todos os alunos saem correndo da sua sala para chegar o mais cedo possível na cantina, tanta é a vontade de comer. Um de seus professores notou, porém, que havia ali uma oportunidade.

Utilizando um sistema de recompensa, seu professor de matemática disse que a ordem da fila para se servir será dada não pela ordem de chegada, mas sim pela soma das notas obtidas em sala de aula. Assim, aqueles com maior nota poderão se servir antes daqueles que tem menor nota.

Sua tarefa é simples: dada a ordem de chegada dos alunos na cantina, e as suas respectivas notas na matéria de matemática, reordene a fila de acordo com as notas de matemática, e diga quantos alunos não precisaram trocar de lugar nessa reordenação.

Entrada

A primeira linha contém um inteiro N , indicando o número de casos de teste a seguir.

Cada caso de teste inicia com um inteiro M ($1 \leq M \leq 1000$), indicando o número de alunos. Em seguida haverá M inteiros distintos P_i ($1 \leq P_i \leq 1000$), onde o i -ésimo inteiro indica a nota do i -ésimo aluno.

Os inteiros acima são dados em ordem de chegada, ou seja, o primeiro inteiro diz respeito ao primeiro aluno a chegar na fila, o segundo inteiro diz respeito ao segundo aluno, e assim sucessivamente.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, indicando o número de alunos que não precisaram trocar de lugar mesmo após a fila ser reordenada.

Exemplo

Entrada	Saída
3	1
3	0
100 80 90	4
4	
100 120 30 50	
4	
100 90 30 25	

Problema E: Gerando Permutações Ordenadas Rapidamente

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1401>

Gerar permutações sempre foi um problema importante na ciência da computação. Neste problema, você terá de gerar todas as permutações de uma dada string, em ordem lexicográfica crescente. Lembre-se que seu algoritmo deve ser eficiente.

Entrada

A primeira linha da entrada contém um inteiro n , indicando o número de strings que seguem. As próximas n linhas contém uma string cada. Cada string conterá apenas caracteres alfanuméricos, e nunca conterá espaços. O tamanho máximo de uma string é 10.

Saída

Para cada string da entrada, imprima todas as permutações possíveis da string, em ordem lexicográfica crescente. Note que as strings devem ser tratadas como Case Sensitive (isto é, letras maiúsculas são diferentes das minúsculas). Além disso, nenhuma permutação deve ser impressa mais de uma vez. Uma linha em branco deve ser impressa após cada lista de permutações.

Exemplo

Entrada	Saída
3 ab abc bca	ab ba abc acb bac bca cab cba abc acb bac bca cab cba

Problema F: Estiagem

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1023>

Devido às constantes estiagens que aconteceram nos últimos tempos em algumas regiões do Brasil, o governo federal criou um órgão para a avaliação do consumo destas regiões com finalidade de verificar o comportamento da população na época de racionamento. Este órgão responsável irá pegar algumas cidades (por amostragem) e verificará como está sendo o consumo de cada uma das pessoas da cidade e o consumo médio de cada cidade por habitante.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro N ($1 \leq N \leq 1 \cdot 10^6$), indicando a quantidade de imóveis. As N linhas contém um par de valores X ($1 \leq X \leq 10$) e Y ($1 \leq Y \leq 200$), indicando a quantidade de moradores de cada imóvel e o respectivo consumo total de cada imóvel (em m^3). Com certeza, nenhuma residência consome mais do que $200 m^3$ por mês. O final da entrada é representado pelo número zero.

Saída

Para cada entrada, deve-se apresentar a mensagem “Cidade# n:”, onde n é o número da cidade seguindo a sequência (1, 2, 3, ...) e em seguida deve-se listar, por ordem ascendente de consumo, a quantidade de pessoas seguido de um hífen e o consumo destas pessoas, arredondando o valor para baixo. Na terceira linha da saída deve-se mostrar o consumo médio por pessoa da cidade, com 2 casas decimais sem arredondamento, considerando o consumo real total. Imprimir uma linha em branco entre dois casos de testes consecutivos. No fim da saída não deve haver uma linha em branco.

Exemplo

Entrada	Saída
3 3 22 2 11 3 39 5	Cidade# 1: 2-5 3-7 3-13 Consumo medio: 9.00 m3.
1 25 2 20 3 31 2 40 6 70 0	Cidade# 2: 5-10 6-11 2-20 1-25 Consumo medio: 13.28 m3.

Problema G: Bolhas e Baldes

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1023>

Andrea, Carlos e Marcelo são muito amigos e passam todos os finais de semana à beira da piscina. Enquanto Andrea se bronzeia ao sol, os dois ficam jogando Bolhas. Andrea, uma cientista da computação muito esperta, já disse a eles que não entende por que passam tanto tempo jogando um jogo tão primário.

Usando o computador portátil dela, os dois geram um inteiro aleatório N e uma sequência de inteiros, também aleatória, que é uma permutação de $1, 2, \dots, N$.

O jogo então começa, cada jogador faz um movimento, e a jogada passa para o outro jogador. Marcelo é sempre o primeiro a começar a jogar. Um movimento de um jogador consiste na escolha de um par de elementos consecutivos da sequência que estejam fora de ordem e em inverter a ordem dos dois elementos. Por exemplo, dada a sequência 1, 5, 3, 4, 2, o jogador pode inverter as posições de 5 e 3 ou de 4 e 2, mas não pode inverter as posições de 3 e 4, nem de 5 e 2. Continuando com o exemplo, se o jogador decide inverter as posições de 5 e 3 então a nova sequência será 1, 3, 5, 4, 2. Mais cedo ou mais tarde, a sequência ficará ordenada. Perde o jogador impossibilitado de fazer um movimento. Andrea, com algum desdém, sempre diz que seria mais simples jogar cara ou coroa, com o mesmo efeito. Sua missão, caso decida aceitá-la, é determinar quem ganha o jogo, dada a sequência inicial.

Entrada

A entrada contém vários casos de teste. Os dados de cada caso de teste estão numa única linha, e são inteiros separados por um espaço em branco. Cada linha contém um inteiro N ($2 \leq N \leq 10^5$), seguido da sequência inicial $P = (X_1, X_2, \dots, X_N)$ de N inteiros distintos dois a dois, onde $1 \leq X_i \leq N$ para $1 \leq i \leq N$.

O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha, com o nome do vencedor, igual a **Carlos** ou **Marcelo**, sem espaços em branco.

Exemplo

Entrada	Saída
5 1 5 3 4 2	Marcelo
5 5 1 3 4 2	Carlos
5 1 2 3 4 5	Carlos
6 3 5 2 1 4 6	Carlos
5 5 4 3 2 1	Carlos
6 6 5 4 3 2 1	Marcelo
0	

Problema H: Diga-me a Frequência

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1251>

Dada uma linha de texto, você deve encontrar as frequências de cada um dos caracteres presentes nela. As linhas fornecidas não conterão nenhum dos primeiros 32 ou dos últimos 128 caracteres da tabela ASCII. É claro que não estamos levando em conta o caractere de fim de linha.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é composto por uma única linha de texto com até 1000 caracteres.

Saída

Imprima o valor ASCII de todos os caracteres presentes e a sua frequência de acordo com o formato abaixo. Uma linha em branco deverá separar 2 conjuntos de saída. Imprima os caracteres ASCII em ordem ascendente de frequência. Se dois caracteres estiverem presentes com a mesma quantidade de frequência, imprima primeiro o caractere que tem valor ASCII maior.

Exemplo

Entrada	Saída
AAABBC	67 1
122333	66 2
	65 3
	49 1
	50 2
	51 3

Problema I: 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