

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 11

(Estrutura de Dados: Dicionário)

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: Ida à Feira

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1050>

Observação: Compare a solução usando dicionário com a sua solução realizada na aula 3.

Leia um número inteiro que representa um código de DDD para discagem interurbana. Em seguida, informe à qual cidade o DDD pertence, considerando a tabela abaixo:

DDD	Destination
61	Brasilia
71	Salvador
11	Sao Paulo
21	Rio de Janeiro
32	Juiz de Fora
19	Campinas
27	Vitoria
31	Belo Horizonte

Se a entrada for qualquer outro DDD que não esteja presente na tabela acima, o programa deverá informar: DDD não cadastrado.

Entrada

A entrada consiste de um único valor inteiro.

Saída

Imprima o nome da cidade correspondente ao DDD existente na entrada. Imprima *DDD nao cadastrado* caso não existir DDD correspondente ao número digitado.

Exemplo

Entrada	Saída
11	Sao Paulo

Problema B: Ajude Girafales

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1911>

Observação: Compare a solução usando dicionário com a sua solução realizada na aula 4.

Minutos antes do término das aulas, professor Girafales passa uma lista de presença. Certo dia, ele resolveu conferir as assinaturas e notou que alguns alunos assinavam diferente em algumas aulas e desconfiou que alguém poderia estar assinando por eles. Como o professor possui muitos alunos e pouco tempo (o café com dona Florinda é prioridade), ele pediu sua ajuda para validar as assinaturas. Uma assinatura é considerada falsa se houver mais de uma diferença entre a original e a que estiver sendo checada. Considere diferença uma troca de maiúscula para minúscula ou o contrário.

Entrada

Haverá diversos casos de testes. A primeira linha de cada caso inicia com um inteiro N ($1 \leq N \leq 50$) representando a quantidade de alunos de sua turma. As próximas N linhas serão da seguinte forma:

Nome do aluno Assinatura Original

A seguir haverá um inteiro M ($0 \leq M \leq N$), representando a quantidade de alunos que compareceram a uma aula. M linhas seguem, no seguinte formato:

Nome do aluno Assinatura na aula

Todos os alunos possuem apenas o primeiro nome na lista, nenhum nome se repete e todos os nomes contêm no máximo 20 letras (**a-z A-Z**).

A entrada termina com $N = 0$, a qual não deve ser processada.

Saída

Para cada caso, exiba uma única linha, a quantidade de assinaturas falsas encontradas.

Exemplo

Entrada	Saída
4	1
Chaves ChAvEs	2
Kiko kikO	
Nhonho NHONHO	
Chiquinha CHIquinHa	
3	
Chaves ChAvEs	
Kiko kIKO	
Chiquinha CHIquinHA	
2	
Jadson jadSON	
Crishna Crishna	
2	
Crishna CRISHNA	
Jadson JADson	
0	

Problema C: O Fantástico Jaspion

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1449>

Em 1985 estreia na TV Japonesa a série Kyojiu Tokusou Jaspion (Investigador Especial de Monstros Jaspion). A série chega ao Brasil alguns anos depois com o título “O Fantástico Jaspion”, e com ela nasce a fantasia de polícia espacial em milhões de brasileiros. As crianças saíam da escola, corriam pelas ruas (sem olhar se vinha carro), ligavam a TV e mergulhavam na coragem, exemplo de pessoa, e incontestável sede por justiça do Fantástico Jaspion. O comércio de gibis e as brigas por figurinhas no recreio da escola estavam alcançando números históricos. Até então, tal sentimento só havia sido estimulado com tanta intensidade pelo Chaves e a sua turma! Diante dessa febre intergaláctica, o inevitável aconteceu. Os produtores do Jaspion ganharam o Nobel da Paz! Isso mesmo! Os produtores ganharam um Nobel. As histórias do grandioso Jaspion estavam por todo canto. Agora as crianças tinham um belíssimo exemplo para seguir. A paz mundial estava garantida! Não precisávamos mais temer o monstrengo Satan Gos!

No Brasil havia uma criança que adorava as histórias do Jaspion! Antônio Melhorança Capote Valente Junior carinhosamente apelidado de ACM, um menino da zona sul de São Paulo que adorava cantar as músicas do grande herói. Ele era tão fanático que chegou a comprar um dicionário de Japonês-Português e iniciou um trabalho árduo de tradução. Entretanto, o trabalho ficou inacabado! Alguns trechos da canção ainda precisam ser traduzidos. Neste momento você deve estar se perguntando: qual é a minha tarefa neste fabuloso problema? Ok! Antes de falar da sua tarefa, convide seu companheiro de equipe para mergulhar com você no desfecho da história. Para isso, vamos falar mais um pouco sobre o nosso ACM. Ele se formou em Ciência da Computação e hoje trabalha no mesmo escritório que você. Pois é! Você trabalha como programador ao lado dessa figura! Como sabemos que você gosta muito dele, temos certeza que vai aceitar a seguinte tarefa: dado um dicionário Japonês-Português e uma letra de música, escreva um programa que imprima a letra traduzida.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias.

A primeira linha de cada instância contém dois inteiros M e N ($1 \leq M \leq 1000000$, $1 \leq N \leq 1000$), que representam o número de palavras no dicionário e o número de linhas na letra da música, respectivamente.

Os próximos M pares de linhas contêm as traduções: a primeira linha de cada par contém a palavra em Japonês, e a segunda linha contém a tradução para o Português (que pode ter uma ou mais palavras). Todas as palavras usam apenas letras minúsculas. Cada palavra em Japonês aparece apenas uma vez em cada instância.

As próximas N linhas contêm a letra da música. Cada linha da letra da música é uma lista de palavras separadas por um espaço (todas as palavras consistem apenas de letras minúsculas). Algumas podem estar vazias, mas nenhuma linha possui espaços no início ou no final.

Nenhuma linha contém mais do que 80 letras.

Saída

Para cada instância imprima as N linhas traduzidas. As palavras que não estão no dicionário devem ser impressas como aparecem na entrada. Imprima uma linha em branco após tradução, inclusive após a última.

Nenhuma linha da saída contém mais do que 80 letras.

Exemplo

Entrada	Saída
2	o cara tossiu
4 3	o cara tossiu
galaxy	o canalha do alto que util
cara tossiu	
kagayaku	o livro esta sobre a mesa
canalha do	
atsuki	
alto que	
yuushi	
util	
o galaxy	
o galaxy	
o kagayaku atsuki yuushi	
3 1	
bashulhan	
sobre a mesa	
hu	
esta	
hasefer	
o livro	
hasefer hu bashulhan	

Problema D: Gloud Computing

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1452>

A Gloud Computing está vindo se instalar para a região de Joinville. Eles são conhecidos por proverem aplicativos na internet, mais especificamente um modelo de negócios baseado em cloud computing - computação nas nuvens.

A fim de selecionar os novos funcionários da empresa, eles contactaram o comitê da maratona da UDESC, para que passassem um problema aos nossos maratonistas. Aquele que resolver, além do balão, pode preencher a ficha funcional com estrelinhas a mais.

Basicamente, a Gloud Computing tem aplicações espalhadas em seus servidores em diversos lugares do mundo. Estes servidores são especializados em uma lista de aplicativos a serem usados pelos usuários ali conectados na internet das nuvens.

Por exemplo, o servidor de Joinville pode disponibilizar a aplicação **A**, enquanto que o de Pasadena na Califórnia provê as aplicações **A**, **B** e **C** e o servidor de Pomerode provê a aplicação **C**.

Temos um conjunto de servidores a cada um com um conjunto de aplicações a serem disponibilizados a um conjunto de usuários. Cada usuário pode estar conectado a um ou mais servidores dependendo de sua demanda, como ilustrado na Figura 1.

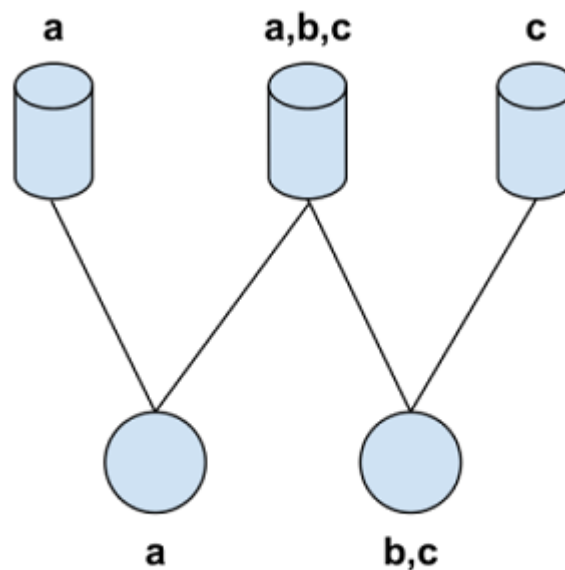


Figura 1: 3 provedores de serviços, 2 usuários e 4 conexões.

Serão disponibilizados a você dados sobre estes dois conjuntos, servidores e demanda dos usuários, e você deverá dizer a quantidade total de conexões entre clientes e servidores. As conexões são feitas de forma a maximizar a redundância. Por exemplo, se um cliente utilizar as aplicações **B** e **C**, ele irá se conectar a todos os servidores que disponibilizarem ao menos a aplicação **B** e a todos os que disponibilizarem ao menos a **C**. Múltiplas conexões entre um mesmo par de cliente e servidor são contabilizadas como um só. Pode ser que um cliente requeira uma aplicação inexistente, assim como o caso de um servidor prover uma aplicação não requisitada por nenhum cliente.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste é iniciado por dois inteiros, N e M ($0 \leq N, M \leq 200$), que correspondem ao número de servidores e ao número de clientes. Cada uma das próximas N linhas contém um valor Q_i ($0 \leq Q_i \leq 100$) correspondente ao número de aplicações fornecidas pelo i -ésimo servidor, seguido por Q_i palavras (separadas por espaços) referentes aos nomes das aplicações fornecidas. Após esta descrição dos servidores, seguem M linhas, cada uma contendo um valor P_j ($0 \leq P_j \leq 100$) correspondente ao número de aplicações requisitadas pelo j -ésimo cliente, seguido por P_j palavras (separadas por espaços) referentes aos nomes das aplicações requisitadas. A entrada termina quando $N = M = 0$. Todos os nomes de aplicativos tem tamanho entre 1 e 20 caracteres.

Saída

Para cada caso de teste, o programa deve imprimir a soma total de conexões entre cliente e servidores em uma linha, desconsiderando múltiplas conexões entre um mesmo par de cliente e servidor.

Exemplo

Entrada	Saída
3 2	4
1 a	4
3 a b c	
1 c	
1 a	
2 b c	
5 2	
2 s1 s2	
2 s3 s4	
2 s5 s6	
2 s7 s8	
2 s1 s2	
3 s1 s2 s20	
3 s1 s2 s21	
0 0	

Problema E: MacPRONALTS

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1985>

O MacPRONALTS está com uma super promoção, exclusivo para os competidores da primeira Seletiva do MaratonaTEC. Só que teve um problema, todos os maratonistas foram tentar comprar ao mesmo tempo, com isso gerou uma fila muito grande. O pior é que a moça do caixa estava sem calculadora ou um programa para ajudá-la a calcular com maior agilidade, eis que surge você para fazer um programa para ajudar a coitada e aumentar a renda do MacPRONALTS. Segue o cardápio do dia contendo o número do produto e seu respectivo valor.

1001 | R\$ 1.50

1002 | R\$ 2.50

1003 | R\$ 3.50

1004 | R\$ 4.50

1005 | R\$ 5.50

Entrada

A primeira entrada informada é a quantidade de produtos comprados ($1 \leq p \leq 5$). Para cada produto segue a quantidade ($1 \leq q \leq 500$) que o consumidor comprou.

Obs.: não poderão ser informados números de produtos repetidos.

Saída

Você deve imprimir o valor da compra com duas casas decimais.

Exemplo

Entrada	Saída
2 1001 2 1005 3	19.50

Entrada	Saída
1 1003 500	1750.00

Entrada	Saída
5 1001 500 1005 300 1003 23 1002 52 1004 44	2808.50

Problema F: O Campeão

<https://www.urionlinejudge.com.br/judge/pt/problems/view/2040>



Thyago é um torcedor fanático do Sport Clube de Recife e sempre acompanha as finais dos campeonatos em que seu time está jogando. Como os jogos são geralmente à noite e Thyago está ocupado nesse horário trabalhando ou ministrando aula, ele não está com tempo de ver os jogos do seu time de coração e nem acompanhar a pontuação total do campeonato ao longo do dia, no qual está na rodada final. Deoclécio, que é amigo de Thyago e também gosta muito de futebol, está assistindo o campeonato e anotou a pontuação de todos os times até então e também o placar dos jogos da rodada final.

Sua tarefa é ajudar Deoclécio fazendo um programa que, dado o nome dos times participantes do campeonato, a sua pontuação até então e o placar dos últimos jogos, determine qual foi o time campeão e a pontuação final desse time.

Regras desse campeonato:

- Cada gol marcado valerá 3 pontos;
- Caso os times empatem, será adicionado +1 ponto para cada time da partida;
- O time vencedor de uma partida ganhará +5 pontos.

Observações:

- É garantido que sempre haverá um time vencedor do campeonato;
- O time Sport sempre aparece nas entradas do problema;
- Não haverá empates de pontuação final entre dois ou mais times após os placares dados;
- O nome do time da entrada sempre terá apenas uma palavra;
- Os nomes dos times não se repetem e não contém caracteres especiais;
- Nenhum dos times joga mais de uma vez (mais de uma partida no mesmo caso de teste).

Entrada

A entrada contém vários casos de teste. A primeira linha contém um inteiro N ($2 \leq N \leq 100$) representando a quantidade de times a seguir que estão participando do campeonato. Seguem N_i linhas contendo o nome do time **S** e a pontuação do time **P** até então ($0 \leq P \leq 100$). Logo após, seguem $N/2$ linhas contendo o placar de cada jogo da última rodada, no formato "**TimeA** gols**A**-gols**B** **TimeB**", conforme ilustrado no exemplo abaixo. A entrada termina com $N = 0$, na qual não deve ser processada.

Saída

Para cada caso de teste na entrada, seu programa deve produzir uma linha de saída, contendo caso o Sport seja o campeão: "O Sport foi o campeao com **X** pontos :D", sendo **X** a quantidade total de pontos. Caso contrário, o seu programa deve produzir a

seguinte linha: "O Sport nao foi o campeão. O time campeão foi o **S** com **X** pontos :(", sendo **S** o nome do time vencedor e **X** a quantidade total de pontos. Deixe uma linha em branco depois de cada caso de teste.

Exemplo

Entrada	Saída
4 Treze 20 Campinense 35 Fortaleza 25 Sport 20 Sport 4-0 Campinense Fortaleza 2-2 Treze 6 Bahia 42 Sport 43 Vitoria 41 Fortaleza 32 Ceara 33 Campinense 22 Sport 2-0 Bahia Vitoria 3-1 Ceara Campinense 2-1 Fortaleza 0	O Sport foi o campeão com 37 pontos :D O Sport nao foi o campeão. O time campeão foi o Vitoria com 55 pontos :(