

Problema I

Trilhos do Trem

Arquivo fonte: `trem.{c | cpp | java}`

Autor: Anderson Viçoso de Araújo (UFMS)

A empresa de trens e metrô *NewTrainBraza* ganhou uma licitação para construir estações e trilhos em cidades de grande porte do Brasil com o intuito de reduzir o fluxo de carros pelas ruas e melhorar a qualidade de vida dos seus habitantes. Para isso, a empresa pediu para que seus técnicos definissem os tipos de trilhos (reto, curvo, na superfície, subterrâneo etc.) que serão instalados entre as estações das cidades.

Para tanto, os técnicos decidiram utilizar uma cadeia de caracteres que representa as estações e os tipos de trilhos existentes entre elas. O mapa dos trens da cidade *FooBar*, por exemplo, é definido por meio da cadeia de caracteres: "A 12 - ~ 7 _ B ~ 4 # C" (desconsiderando as aspas), onde todos os caracteres são separados por um espaço em branco e o significado de cada um deles é apresentado na Tabela 1. De acordo com esta cadeia de caracteres, entre a estação A e B, existem 12 trilhos retos (na superfície), 1 trilho em curva e 7 trilhos retos subterrâneos.

É importante notar que, em cada tipo de trecho, o trem desenvolve uma velocidade constante proporcional à sua velocidade de entrada. Desta forma, caso o trem saia a 100 km/h da estação A, ele percorrerá o trecho com 12 trilhos retos (na superfície) na mesma velocidade (de acordo com a Tabela 1, 100% da velocidade de entrada). Em seguida, ele percorrerá um trecho em curva com uma velocidade média de 70 km/h (70% da velocidade de entrada). E assim por diante.

Você, um especialista em computação, vai ajudar a empresa a construir um sistema para calcular o tempo que os seus trens demoram para realizar o trajeto entre uma estação e outra.

Tabela 4. Descrição das informações da cadeia de caracteres que representa o mapa dos trens.

Caracteres	Descrição	Função	Velocidade
-	Hífen	Trecho de trilho em reta na superfície	100%
_	Sublinhado (<i>Underline</i>)	Trecho de trilho em reta subterrânea	110%
~	Til	Trecho de trilho em curva	70%
#	Cerquilha	Trecho de trilho em superfície sobre ponte	80%
A-Z	Alfabeto (maiúsculas)	Indicação de uma estação	-
2-999	Número inteiro	Número de trechos com mesma característica	-

Entrada

A entrada é composta por vários conjuntos de casos de teste. A primeira linha de cada conjunto de casos de teste contém uma cadeia de caracteres *M* (entre 1 e 512 caracteres), correspondendo ao mapa dos trilhos de uma cidade qualquer. A cadeia *M* deve iniciar e terminar com uma letra maiúscula entre A e Z (que corresponde a uma estação do trem). Entre quaisquer letras maiúsculas, existe uma combinação dos caracteres descritos na Tabela 1, separados por espaços em branco. Nas combinações, é possível ter um número (entre 2 e 999) indicando o número de trilhos de mesmo tipo em sequência. A próxima linha indica três números inteiros *T*, *V* e *N* ($1 \leq T \leq 1000$, $1 \leq V \leq 1000$ e $1 \leq N \leq 1000$), separados por um espaço em branco, onde *T* representa o valor em metros correspondente a um trilho do percurso; *V* indica a velocidade

constante do trem em km/h em reta na superfície (100% na Tabela 1) e N o número de casos de teste a serem analisados. Seguem-se N linhas com dois caracteres cada, separados por um espaço em branco, X e Y (ambos com algum caractere entre A e Z), indicando a estação de partida e de chegada para as quais deve-se calcular o tempo de viagem, respectivamente. As entradas terminam com o fim do arquivo.

Saída

Para cada caso de teste lido, escreva o texto “Cidade X” onde X indica o número do conjunto de casos de teste associado a um mapa de trilhos. Para cada caso de teste, escreva uma linha com o tempo que o trem vai levar para chegar entre as estações definidas no teste. O formato da saída para o tempo deve estar de acordo com a *string*: **HHh MMmin SSs**, onde as maiúsculas *HH* ($00 \leq HH \leq 99$), *MM* e *SS* ($00 \leq MM$ e $SS \leq 59$) indicam os números inteiros correspondentes às horas, minutos e segundos, respectivamente. **IMPORTANTE: valores com precisão de tempo menores que segundos (centésimos) devem ser truncados.** Por fim, uma linha em branco deve ser deixada após cada conjunto de casos de teste.

Exemplos

Entrada:	Saída:
A 12 - ~ 7 _ B ~ 4 # C 100 72 2 A B A C A 84 ~ - # B ~ _ 26 - C ~ - 26 ~ # D - 12 ~ 12 - _ E 85 ~ 23 - F 29 - 3 _ - G - 62 ~ # 44 _ 24 ~ _ H 761 73 1 E F A 32 # - _ ~ B 93 ~ _ 24 # 45 ~ _ ~ 56 _ 59 ~ C 80 160 2 A C B C	Cidade 1: 00h 01min 38s 00h 02min 11s Cidade 2: 01h 30min 20s Cidade 3: 00h 12min 16s 00h 10min 58s

Solução

Este é um problema considerado de dificuldade média, usa física (cálculo do tempo) e manipulação de strings.

Podemos chegar a solução deste problema através dos passos:

1. Para cada string de trechos lida, incrementar um contador de cidades e imprimir o rótulo da cidade corrente;
2. Depois de ler as entradas, converter a velocidade de entrada para m/s;
 - a. Para cada pesquisa de percurso entre estações, percorrer a string do caminho, retirando os espaços
 - i. Procurar a letra correspondente a estação de início para iniciar o processamento dos caracteres;
 - ii. Verificar se o token é um número, se sim, armazenar para multiplicação posterior;
 - iii. Caso o token seja um dos caracteres indicados na tabela (-,_,~,#) verificar a porcentagem da velocidade correspondente que deve ser considerada, multiplicar pelo número anterior (caso exista), multiplicar pela quantidade de metros por trecho e dividir pela velocidade no trecho. Somar os resultados, que vai corresponder ao tempo em segundos;
 - iv. Ao encontrar a letra correspondente a estação final, parar o processamento da cadeia;
 - v. Transformar o tempo em segundos para horas, minutos e segundos usando resto e divisão por 60;
 - vi. Imprimir a string do tempo.