# Programadores Devem Aprender Teoria da Computação

Por Matheus Pimenta, UnB Brazil

**Timelimit: 1**

---

Durante sua aventura em Imaginalândia, Alan Leopold "Butters" Stotch Turing inventou sua famosa máquina de fazer sorvete. Basta dizer para a máquina o sabor, que ela faz um sorvete delicioso!

Fonte da imagem: http://southpark.cc.com/avatar

Neste momento, Butters está preocupado com uma coisa. Ele é capaz de construir sua máquina de muitas maneiras diferentes; e está fazendo experimentos para determinar qual é a melhor. Você decidiu ajudá-lo, pois está ansioso por um sorvete de creme. Dada a descrição de uma máquina e uma série de **Q** consultas de sabores, Butters quer saber quantos *passos* esta máquina leva para fabricar o sorvete de cada consulta.

Uma máquina de sorvete é uma configuração com um estado (um número inteiro), uma cadeia e uma posição nesta cadeia. Para cada configuração <estado,cadeia,posição>, um *passo* é gerar uma nova configuração: atualiza-se o estado, atualiza-se o símbolo que está na posição atual; e move-se a posição atual para uma posição adjacente (à esquerda, ou à direita). Caso o comando movimente a posição para uma posição além dos limites da cadeia, deve ser concatenado um espaço em branco no respectivo extremo; e a posição da nova configuração deve apontar para este espaço em branco. A máquina começa na configuração <1,**sabor**,1>, onde **sabor** é uma cadeia e o segundo 1 indica a primeira posição desta cadeia. A máquina termina de fazer o sorvete quando atinge uma configuração cujo estado é o inteiro **S**, de sorvete.

O truque mágico é que, para cada configuração <estado,cadeia,posição>, a *máquina de Butters* é capaz de executar vários passos distintos, de modo que ela pode terminar de fazer o sorvete mais rapidamente. Sempre que a máquina chega a uma configuração que leva a múltiplas novas configurações, a máquina cria cópias de si mesma, de modo que cada cópia segue independentemente. Há uma nova cópia para cada nova configuração. Após gerar as cópias para as novas configurações, a máquina morre. Caso uma configuração não gere novas cópias, ela só morre. O processo termina quando alguma cópia termina de fazer o sorvete. É garantido que alguma ramificação da máquina terminará de fazer o sorvete.

## Entrada

A entrada é composta por vários casos de teste e termina com fim de arquivo.

A primeira linha de um caso de teste contém os inteiros **N**, **S** e **Q**, onde $0 \leq \mathbf{N} \leq 25$ e $1 \leq \mathbf{S}, \mathbf{Q} \leq 10$.

As próximas **N** linhas descrevem os comandos da máquina a ser testada. Cada linha está no formato **q a t b c**, indicando que se uma configuração estiver no estado **q** e o símbolo na posição atual for **a**, então deve-se gerar uma nova configuração com estado **t**, atualizar o símbolo na posição atual para **b** e deve-se mover a posição na direção **c**, de acordo com a descrição do enunciado. Note que $1 \leq \mathbf{q}, \mathbf{t} \leq \mathbf{S}$. O dado **a** pode ser uma letra minúscula, '0', ou '~' seguido de uma cadeia não-vazia **w**, que pode conter letras minúsculas ou '0'. No terceiro caso, o comando deve ser executado quando o símbolo na posição atual não aparecer em **w**. O dado **b** pode ser uma letra minúscula, '0', ou '*'. No terceiro caso, o símbolo na posição atual não deve ser atualizado. O dado **c** vale 'E' (esquerda), ou 'D' (direita). O símbolo '0' significa espaço em branco.

As próximas **Q** linhas descrevem as consultas. Cada linha é uma cadeia **sabor** de letras minúsculas, com no mínimo 1 e no máximo 20 letras.

## Saída

Para cada consulta de cada caso de teste, imprima uma linha com o número de passos da ramificação que produziu o sorvete.

| Exemplo de Entrada | Exemplo de Saída |
|---|---|
| 0 1 3<br>chocolate<br>morango<br>baunilha<br>9 9 2<br>1 c 2 * D<br>2 a 3 * D<br>3 f 4 * D<br>4 e 9 * D<br>1 c 5 * D<br>5 a 6 * D<br>6 c 7 * D<br>7 a 8 * D<br>8 u 9 * D<br>cafe<br>cacau<br>11 6 3<br>1 ~a0 1 * D<br>1 a 2 0 D<br>2 ~0 2 * D<br>2 0 3 * D<br>3 ~0 3 * D<br>3 0 4 a E<br>4 ~0 4 * E<br>4 0 5 * E<br>5 ~0 5 * E<br>5 0 1 a D<br>1 0 6 * D<br>chocolate | 0<br>0<br>0<br>4<br>5<br>18<br>18<br>31 |

| Exemplo de Entrada | Exemplo de Saída |
| --- | --- |
| morango<br>baunilha | |

Perceba que a solução deste problema é um interpretador!