

Indentador de BF

| | |
|---------------------|-----------------------------|
| Arquivo de entrada: | <code>entrada padrão</code> |
| Arquivo de saída: | <code>saída padrão</code> |
| Tempo limite: | 1 segundo |
| Limite de memória: | 256 megabytes |

A linguagem brainf*ck é uma linguagem de programação esotérica desenhada para desafiar e divertir programadores. Ela foi criada em 1993 por Urban Müller, e pode ser usada para descrever uma família de máquinas de Turing. A linguagem é executada usando uma fita unidimensional de octetos (chamados de células) e um ponteiro para esta fita. A linguagem possui oito comandos:

- ‘>’ incrementa o ponteiro (move o ponteiro para a direita)
- ‘<’ decrementa o ponteiro (move o ponteiro para a esquerda)
- ‘+’ incrementa a célula apontada pelo ponteiro
- ‘-’ decrementa a célula apontada pelo ponteiro
- ‘.’ imprime na tela o valor da célula apontada pelo ponteiro
- ‘,’ lê da entrada para a célula apontada pelo ponteiro
- ‘[’ se a célula apontada pelo ponteiro for zero, pula para o ‘]’ correspondente
- ‘]’ volta para o ‘[’ correspondente

Quaisquer outros caracteres são ignorados, o que permite que o código brainf*ck seja “comentado”. Você decidiu escrever alguns programas em brainf*ck, mas como os espaços e quebras de linhas são ignorados mesmo, você começou a escrever tudo junto na mesma linha. Para permitir que seu código fique mais “legível” (como se isso fosse possível numa linguagem assim), você decidiu codar um indentador. O funcionamento é simples:

- Se existir pelo menos um caractere não-espaço entre dois comandos, é necessário que o primeiro comando seja seguido de um espaço, do “comentário” e uma quebra de linha, e em seguida, o segundo comando.
- Se existir apenas um caractere de espaço entre dois comandos, este espaço é removido.
- Os comandos ‘[’ e ‘]’ criam um nível de indentação de um espaço apenas se entre os comandos existir pelo menos um caractere ignorado não-espaço. Um nível de indentação significa que todas as linhas entre estes dois comandos devem estar com seu nível de indentação incrementado, e os comandos ‘[’ e ‘]’ ficam isolados em sua própria linha contendo o seu próprio comentário.

Implemente um programa que pega um código-fonte em brainf*ck em uma única linha e devolve o código-fonte indentado.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 10^6$), que determina o tamanho do código-fonte brainf*ck. A segunda linha é código-fonte que tem exatamente N caracteres que podem ser comandos, espaços, números e letras latinas minúsculas. É garantido que o primeiro caractere é um comando. O código-fonte é bem formado em relação aos espaços: Não possui espaços no começo, no final, nem sequências de mais de um espaço.

Saída

Devem ser impresso o código-fonte indentado. Cada linha deve ser composta de um número C ($0 \leq C$), o nível de indentação, um espaço, e o texto daquela linha.

Exemplos

| entrada padrão |
|---|
| 70 ++c0 eh 2>+++++c1 eh 5[<+adiciona 1 a c0-subtrai 1 de c1] <.imprime c0 |
| saída padrão |
| 0 ++ c0 eh 2 0 >+++++ c1 eh 5 0 [1 <+ adiciona 1 a c0 1 - subtrai 1 de c1 0] 0 <. imprime c0 |
| entrada padrão |
| 72 >>c2[-]zera c2<<c0[enquanto c0-subtrai>>c2+incrementa c2<<c0]fim do laco |
| saída padrão |
| 0 >> c2 0 [-] zera c2 0 << c0 0 [enquanto c0 1 - subtrai 1 >> c2 1 + incrementa c2 1 << c0 0] fim do laco |
| entrada padrão |
| 47 [[[+laco infinito dentro do laco infinito]-[]] |
| saída padrão |
| 0 [1 []+ 1 [laco infinito dentro do laco infinito 1] 1 -[] 0] |