Brainfuck

Linguagens de programação, times de futebol e religião não se discutem. Cada um tem seus favoritos e não admite que o do outro seja melhor (que me perdoem os corinthianos, palmeirenses e são-paulinos). Um grupo de pesquisadores (que não tinha o que fazer) resolveu escrever uma linguagem de programação ``ideal'' (ideal prá quem, cara pálida?): o Brainfuck.

Brainfuck é uma linguagem de programação cujo funcionamento é muito parecido com uma máquina de Turing. Essa máquina possui como componentes um vetor de 30000 bytes, indexado de 0 a 29999, e um ponteiro, que guarda uma posição desse vetor.

Em cada passo, a máquina realiza uma instrução de acordo com o byte armazenado na posição do vetor indicada pelo ponteiro. Quando esse byte é igual a zero, a execução é terminada.

O conjunto de instruções válidas da linguagem é o seguinte:

Instrução	Descrição
>	Incrementa o ponteiro.
<	Decrementa o ponteiro.
+	Incrementa o byte na posição indicada pelo ponteiro.
-	Decrementa o byte na posição indicada pelo ponteiro.
.	Imprime o valor do byte na posição indicada pelo ponteiro.
,	Lê um byte e armazena na posição indicada pelo ponteiro. Se não houver nada que possa ser lido (entrada acabou), armazenar zero.
[	Início do loop: Executa o código delimitado até que o byte na posição indicada pelo ponteiro seja igual a zero.
]	Fim do loop.
#	Imprime os valores das 10 primeiras posições do vetor.

O ponteiro sempre começa com valor 0, assim como todas as posições do vetor. Na descrição de programas na linguagem brainfuck, caracteres diferentes dos descritos acima são ignorados.

Entrada

A entrada é composta de diversas instâncias. O número de instâncias é dado na primeira linha da entrada. Cada instância começa com uma linha em branco. A próxima linha contém uma cadeia de caracteres não-brancos (ou seja, diferentes de espaço em branco e tabulação), que vai conter a entrada para o programa. Ou seja, os comandos de leitura são realizados nessa cadeia. Toda a entrada para o programa está contida em uma única linha.

Por fim, a terceira linha contém a descrição do programa. Assim como a segunda linha, esta também não contém caracteres brancos e está inteiramente contida em uma única linha (a separação feita no segundo exemplo de entrada foi feita para evitar o estouro de linha).

Tanto a segunda como a terceira linha têm entre 1 e 100000 caracteres.

Saída

Para cada instância, você deverá imprimir um identificador Instancia k, onde h é o número da instância atual. Na linha seguinte você deve imprimir a saída do código fornecido na entrada.

Após cada instância, seu programa deve imprimir uma linha em branco.

Exemplo

Entrada:
2

marrocos
+[>,]<-[+.<-]

nada
++++++++++[>+++++++>++++++++++>+++>+<<<<-]>++.>+.+++++++..+++.>++.<<
+++++++++++++++.>.+++.------.--------.>+.>.

Saída:
Instancia 1
socorram

Instancia 2
Hello World!