

Números e Operações

Por Ginés García Mateos, UM  Espanha

Timelimit: 5

Queremos resolver um problema clássico de obtenção de um determinado número a partir de um conjunto de números iniciais, de acordo com as seguintes regras:

- Todos os números envolvidos no problema são inteiros positivos, e por isso todo resultado intermediário de qualquer operação também deverá ser inteiro. Assim, as operações que levam a não inteiros ou números não positivos serão estritamente proibidas (não vamos considerar truncamento ou resultados de arredondamento, nem qualquer outra coisa do tipo).
- Nós podemos fazer qualquer uma das quatro operações aritméticas elementares: adição, multiplicação, subtração e divisão.
- Cada número pode ser utilizada no máximo uma vez: quando utilizamos dois números, os mesmos não podem ser utilizados novamente em operações posteriores (mas o resultado desta operação pode, e as mesmas considerações aplicam-se a este novo número).
- Não há nenhuma obrigação de fazer uso de todos os números iniciais.

Faremos as seguintes premissas sobre os dados de entrada:

- Para qualquer número de saída T , ele terá sempre que: $0 < T < 2000$.
- Para qualquer número de entrada I , ele terá sempre que: $0 < I < 200$.
- Haverá sempre pelo menos dois números iniciais e sete no máximo.
- Não vamos considerar os casos de problemas triviais, ou seja, casos em que o número de destino já está contido nos iniciais.

A fim de tornar a única solução oferecida pelo programa, também deverá fazer as seguintes suposições sobre a solução que iremos considerar como correta: O programa deverá fornecer a solução que satisfaz as condições, com um número mínimo de operações. Se houver mais do que uma solução com o número mínimo de operações, vamos decidir qual tomar de acordo com as características da primeira operação na qual eles se diferem. As próximas regras serão aplicadas.

- Para duas operações iguais, será escolhido aquele cujo operando do lado esquerdo é maior que, ou igual a, o operando do lado direito. Por exemplo, a operação " $10 \times 5 = 50$ " terá prioridade sobre " $5 \times 10 = 50$ ".
- Para as operações que envolvem dois operadores diferentes, será preferido aquele cujo operador tem a prioridade mais alta, definidos como se segue: Adição (mais alto) - Multiplicação - Subtração - Divisão (mais baixo).
- Para operações com o mesmo operador, será preferido um que envolve o número cuja posição é mais baixa, na ordem em que foram dados na entrada (que aparece em primeiro lugar). Se este primeiro número pode aparecer em duas operações diferentes de mesma precedência, o mesmo critério deve ser aplicado para o segundo operando. Novos números gerados pelas operações são assumidos para serem então, colocados no início da lista ordenada.

Entrada

A entrada é constituída por um conjunto de linhas. Cada linha representa um caso de teste, e lhes serão dados no seguinte formato:

T N I₁ ... I_N

T é o número de destino, **N** é a quantidade de números iniciais e $I_1 \dots I_N$ são os próprios números iniciais. Todos os números são separados por um único espaço em branco. Não haverá caractere (/) levando espaços em branco no início e no fim de qualquer linha de entrada. A linha final com um único zero marca o fim da entrada e não será processada.

Saída

Para cada caso de teste o programa deve produzir uma linha com a mensagem:

No solution

Se o resultado não puder ser obtido a partir dos números de entrada, ou se um conjunto de linhas com a sequência de operações leve a partir dos números iniciais para o resultado, levando em consideração os critérios anteriores. Cada operação deve ser expressa da seguinte forma:

operando_da_esquerda operador operando_da_direita = resultado

Onde *operando_da_esquerda* e *operando_da_direita* representam os números envolvidos na operação, o resultado é o resultado da operação, e o operador é um único caractere que representa a operação que deverá ser feita a partir dos seguintes:

'+' : Adição

'x' : multiplicação

'-' : Subtração

'/' : Divisão

Para cada caso de teste, a última linha de sua solução deve terminar com um símbolo '<-', indicando que esta é a última operação, que dá o número de destino como resultado. Para todos os casos, deverá haver um espaço vazio (apenas um) entre cada impresso e o seguinte. Não deve haver nenhum caractere (/) negociando espaços em branco no início e no fim de cada linha de saída. Cada linha deve terminar com uma marca de final de linha, e não deve aparecer nenhuma linha em branco na saída.

Exemplo de Entrada	Exemplo de Saída
811 6 7 15 19 25 2 3	15 + 7 = 22
911 6 1 2 3 4 5 6	22 x 19 = 418
1897 7 33 2 11 25 7 17 13	418 x 2 = 836
911 7 7 11 23 27 17 7 2	836 - 25 = 811 <-
0	No solution
	25 x 2 = 50
	17 x 13 = 221
	221 + 50 = 271
	271 x 7 = 1897 <-
	23 + 11 = 34
	34 x 27 = 918
	918 - 7 = 911 <-