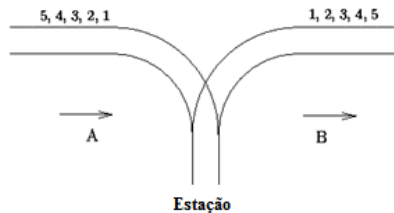


Trilhos

Há uma famosa estação de trem na cidade PopPush. Esta cidade fica em um país incrivelmente acidentado e a estação foi criada no último século. Infelizmente os fundos eram extremamente limitados naquela época. Foi possível construir somente uma pista. Além disso, devido a problemas de espaço, foi feita uma pista apenas até a estação (veja figura abaixo).



A tradição local é que todos os comboios que chegam vindo da direção A continuam na direção B com os vagões reorganizados, de alguma forma. Suponha que o trem que está chegando da direção A tem $N \leq 1000$ vagões numerados **sempre** em ordem crescente **1, 2, ..., N**. O primeiro que chega é o **1** e o último que chega é o **N**. Existe um chefe de reorganizações de trens que quer saber se é possível reorganizar os vagões para que os mesmos saiam na direção B na ordem **$a_1, a_2, a_n...$** .

O chefe pode utilizar qualquer estratégia para obter a saída desejada. No caso do desenho ilustrado acima, por exemplo, basta o chefe deixar todos os vagões entrarem na estação (do 1 ao 5) e depois retirar um a um: retira o 5, retira o 4, retira o 3, retira o 2 e por último retira o 1. Desta forma, se o chefe quer saber se a saída 5,4,3,2,1 é possível em **B**, a resposta seria **Yes**. Vagão que entra na estação **só pode sair para a direção B** e é possível incluir quantos forem necessários para retirar o primeiro vagão desejado.

Entrada

O arquivo de entrada consiste de um bloco de linhas, cada bloco, com exceção do último, descreve um trem e possivelmente mais do que uma requisição de reorganização. Na primeira linha de cada bloco há um inteiro N que é a quantidade de vagões. Em cada uma das próximas linhas de entrada haverá uma permutação dos valores **1, 2, ..., N**. A última linha de cada bloco contém apenas 0. Um bloco iniciando com zero (0) indica o final da entrada.

Saída

O arquivo de saída contém a quantidade de linhas correspondente às linhas com permutações no arquivo de entrada. Cada linha de saída deve ser **Yes** se for possível organizar os vagões da forma solicitada e no caso contrário. Há também uma linha em branco após cada bloco de entrada. No exemplo abaixo, O primeiro caso de teste tem 3 permutações para 5 vagões. O último zero dos testes de entrada não deve ser processados.

Exemplo de Entrada	Exemplo de Saída
5	Yes
5 4 3 2 1	Yes
1 2 3 4 5	No
5 4 1 2 3	
0	Yes
6	
1 3 2 5 4 6	
0	
0	

O Hall dos Assassinos

Quem matou Meryn Trant? Quem matou Syrio Forel, se é que ele de fato morreu? Quem matou Stannis Baratheon? Quem matou Myrcella Baratheon? Quem matou Aerys II Targaryen? Quem vai matar (**alerta de spoiler!**) Jaime Lannister? Para algumas destas perguntas já sabíamos a resposta. Para outras, tínhamos apenas especulações. No entanto, recebemos de um correspondente anônimo uma lista descrevendo vários assassinatos, que já aconteceram ou que estão para acontecer, revelando tanto o nome dos assassinos quanto dos assassinados. Mas os assassinatos não estão em ordem lexicográfica, nem mesmo em ordem cronológica, e fica difícil contar quantas pessoas cada assassino matou. Você pode nos ajudar?

Entrada

Cada linha da entrada descreve um assassinato informando o nome do assassino seguido pelo nome do assassinado. Cada nome é composto por no mínimo um e no máximo 10 caracteres, sendo o primeiro sempre uma letra maiúscula e os demais sempre letras minúsculas. A entrada consiste de no mínimo uma e no máximo 10^5 linhas e é encerrada em *fim de arquivo*.

Saída

A primeira linha da saída deve consistir da frase “HALL OF MURDERERS”, sem as aspas. Cada uma das linhas seguintes deve conter um nome de um assassino seguido do número de pessoas que ele matou. A lista de assassinos deve obedecer a ordem lexicográfica. Se um assassino também acabou sendo assassinado, ele não deve figurar na lista.

Exemplo de Entrada	Exemplo de Saída
Arya Meryn Meryn Syrio Brienne Stannis Ellaria Myrcella Jaime Aerys Brienne Jaime	HALL OF MURDERERS Arya 1 Brienne 2 Ellaria 1