

Infixa para Posfixa em Java

O Professor solicitou que você escreva um programa que converta uma expressão na forma infix (como usualmente conhecemos) para uma expressão na forma posfixa. Como você sabe, os termos in (no meio) e pos (depois) se referem à posição dos operadores. O programa terá que lidar somente com operadores binários +, -, *, /, ^, parênteses, letras e números. Um exemplo seria uma expressão como:

$(A*B+2*C^3)/2*A$. O programa deve converter esta expressão (infixa) para a expressão posfixa: $AB*2C3^{**}+2/A*$

Entrada

A primeira linha da entrada contém um valor inteiro **N** ($N < 1000$), que indica o número de casos de teste. Cada caso de teste a seguir é uma expressão válida na forma infix, com até 300 caracteres.

Saída

Para cada caso, apresente a expressão convertida para a forma posfixa.

Exemplo de Entrada	Exemplo de Saída
3	A2*
A*2	A2*c+d-2/
(A*2+c-d)/2	24*ab^/2c*/
(2*4/a^b)/(2*c)	

Estacionamento Linear em C

Após muito tempo juntando dinheiro, Rafael finalmente conseguiu comprar seu carro (parcelado, é claro). Chega de pegar ônibus, agora sua vida será mais fácil. Pelo menos isso é o que ele pensava, até ouvir falar do estacionamento perto da faculdade onde ele decidiu estacionar o carro todos os dias.

O estacionamento tem apenas um corredor, com largura o suficiente para acomodar um carro, e profundidade suficiente para acomodar **K** carros, um atrás do outro. Como este estacionamento só tem um portão, só é possível entrar e sair por ele.

Quando o primeiro carro entra no estacionamento, o mesmo ocupa a posição próxima à parede, ao fundo do estacionamento. Todos os próximos carros estacionam logo atrás dele, formando uma fila. Obviamente, não é possível que um carro passe por cima de outro, portanto só é possível que um carro saia do estacionamento se ele for o último da fila.

Dados o horário de chegada e saída prevista de **N** motoristas, incluindo Rafael, diga se é possível que todos consigam estacionar e remover seus carros no estacionamento citado.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros **N** e **K** ($3 \leq N \leq 10^4$, $1 \leq K \leq 10^3$), representando o número de motoristas que farão uso do estacionamento, e o número de carros que o estacionamento consegue comportar, respectivamente.

Em seguida haverá **N** linhas, cada uma contendo dois inteiros **C_i** e **S_i** ($1 \leq C_i, S_i \leq 10^5$), representando, respectivamente, o horário de chegada e saída do motorista **i** ($1 \leq i \leq N$). Os valores de **C_i** são dados de forma crescente, ou seja, **C_i** < **C_{i+1}** para todo $1 \leq i < N$.

Não haverá mais de um motorista que chegam ao mesmo tempo, e nem mais de um motorista que saiam ao mesmo tempo. É possível que um motorista consiga estacionar no mesmo momento em que outro motorista deseja sair.

O último caso de teste é indicado quando **N** = **K** = 0, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo a palavra “Sim”, caso seja possível que todos os **N** motoristas façam uso do estacionamento, ou “Nao” caso contrário.

Exemplo de Entrada		Exemplo de Saída	
3 2		Sim	
1 10		Nao	
2 5			
6 9			
3 2			
1 10			
2 5			
6 12			
0 0			