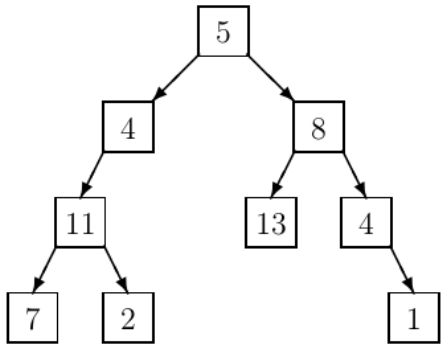


LISP foi uma das primeiras linguagens de programação de alto nível e, com FORTRAN, é uma das linguagens mais antigas atualmente em uso. As listas, que são as estruturas de dados fundamentais em LISP, podem ser facilmente adaptadas para representar outras estruturas de dados importantes, como árvores. Este problema trata e determinar se as árvores binárias representadas como LISP S-expressões possuem uma determinada propriedade.

Dada uma árvore binária de inteiros, você deve escrever um programa que determine se existe um caminho raiz para folha cujos nós somam um inteiro especificado.

Por exemplo, na árvore mostrada em cima, existem exatamente quatro caminhos de raiz para folha. As somas dos caminhos são 27, 22, 26 e 18. As árvores binárias são representadas no arquivo de entrada como LISP S-expressions com o seguinte formato.



árvore-vazia ::= ()
árvore ::= *árvore-vazia* | (*inteiro* *árvore* *árvore*)

A árvore esquematizada acima é representada pela expressão

(5 (4 (11 (7 () ()) (2 () ())) ()) (8 (13 () ()) (4 () (1 () ())) ()))

Observe que, com essa formulação, todas as folhas de uma árvore têm a forma

(*inteiro* () ())

Como uma árvore vazia não possui caminhos raiz para folha, qualquer consulta sobre a existência de um caminho cuja soma seja um inteiro especificado em uma árvore vazia deve ser respondida negativamente.

Entrada

A entrada consiste em uma sequência de casos de teste na forma de pares inteiro/árvore. Cada caso de teste consiste em um número inteiro seguido por um ou mais espaços seguidos por uma árvore binária formatada como uma S-expressão conforme descrito acima. Todas as S-expressões de árvore binária serão válidas, mas as expressões podem ser espalhadas por várias linhas e podem conter espaços. Haverá um ou mais casos de teste em um arquivo de entrada e a entrada é encerrada pelo fim do arquivo.

Saída

Deve haver uma linha de saída para cada caso de teste (par inteiro/árvore) no arquivo de entrada. Para cada par I, T (I representa o inteiro, T representa a árvore) a saída é a string 'yes' se houver um caminho da raiz para a folha em T cuja soma é I e 'no' se não houver caminho em T cuja soma é I.

Entrada de amostra

22 (5(4(11(7()())(2()()))()) (8(13()())(4()(1()()))))
20 (5(4(11(7()())(2()()))()) (8(13()())(4()(1()()))))
10 (3
 (2 (4 () ())
 (8 () ()))
 (1 (6 () ()))
 (4 () ())))
5 ()

Saída de amostra

yes
no
yes
no