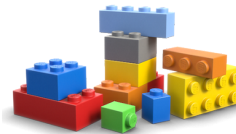


## Problema: Peças de lego

Arquivo: `pecas_lego.c`

Margarida é uma criança que ama brincar com suas peças de lego depois da escola. O grande problema dela é que ela deixa suas peças de lego espalhadas pelo seu quarto, virando um grande campo minado. Margarida já recebeu várias broncas da sua mãe para arrumar seu quarto porque um dia ela iria perder alguma pecinha do lego que ela tanto ama.



Dito e feito, certo dia Margarida se encontra desesperada por não achar o *brick*  $X \times X$ , que é uma peça retangular que possui  $X$  *plates*, sendo *plate* a peça circular do bloco de lego.

Com medo da sua mãe, ajude a Margarida a criar um programa que receba toda a coleção de peças de lego e encontre o *brick*  $X \times X$ . Se mesmo depois de arrumado a peça não estiver lá, é hora de pedir ajuda da mãe. Ela com certeza vai esfregar na cara da Margarida se ela achar.

### Entrada

A primeira linha contém o inteiro  $N$  ( $1 \leq T \leq 100$ ), que representa a quantidade de *brick* que a Margaria possui.

A segunda linha possui  $N$  números  $V_i$  ( $1 \leq V_i \leq 1000000$ ), onde cada número  $V_i$  representa um elemento da coleção de *brick*  $V_i \times V_i$ .

A terceira linha possui um número inteiro  $X$ , que representa o *brick*  $X \times X$  que a Margaria perdeu na bagunça.

### Saída

A saída é composta por exatamente uma linha contendo a palavra **achou** caso a peça  $X$  esteja na coleção da Margarida e **nao achou** caso a peça esteja perdida em outro lugar e a Margarida precise da ajuda da mãe.

Observe os casos de exemplos para melhor entendimento da saída.

### Exemplos

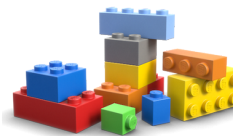
Exemplo de entrada	Saída para o exemplo de entrada
5 3 4 1 2 6 5	nao achou

Exemplo de entrada	Saída para o exemplo de entrada
6 3 5 3 3 1 8 1	achou

## Problema: Lego Bricks

Arquivo: pecas\_lego.c

Daisy is a child who loves to play with her lego bricks after school. The biggest problem is that she usually leaves her lego bricks scattered on the floor, which makes her room a minefield. Daisy already has been scolding by her mother to clean her room because one day Daisy would lose one of her precious lego bricks that she loves so much.



No sooner said than done, one day Daisy was desperate because she had lost the brick  $X \times X$ , which is a rectangular piece that has  $X$  plates.

Since Daisy is afraid of her mother, help her to create a program which receive all of her lego bricks collection and find the brick  $X \times X$ . If even after the clean up the piece continues to be lost, it is time to ask the help of her mother. She probably will scold Daisy again if she finds the lost lego brick.

### Input

The first line contains an integer  $N$  ( $1 \leq T \leq 100$ ), which indicates the amount of lego bricks that Daisy has.

The second line contains  $N$  integers numbers,  $V_i$  ( $1 \leq V_i \leq 1000000$ ), which each  $V_i$  number indicates an element of the *brick*  $V_i \times V_i$  collection.

The third line contains an integer number  $X$ , which indicates the lost *brick*  $X \times X$ .

### Output

The output is only one line which contains the word **achou** to indicates the piece  $X$  was found in Daisy collections and **nao achou** to indicates that the piece was not found and Daisy will needs her mother's help.

For each test case, shows a line with **S** to indicate that  $n$  is a bold prime number, or **N** otherwise.

Check the examples below for better understanding on how the output must be.

### Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
5 3 4 1 2 6 5	nao achou

Exemplo de entrada	Saída para o exemplo de entrada
6 3 5 3 3 1 8 1	achou

*Author: Bruno Ribas - adaptado por Sinayra Moreira*