

É primo ou não é?

Atualmente, os algoritmos mais conhecidos de criptografia assimétrica são baseados em números primos. Esses algoritmos visam cifrar os dados trafegados na rede, para que, caso alguém consiga capturar os dados, eles não possam ser compreendidos. O objetivo é, portanto, que apenas quem possui a chave da criptografia consiga decifrar o que foi enviado e então ler os dados trafegados.

Tais algoritmos só são tidos como seguros porque, atualmente, o tempo para verificar se um dado número N é primo ou não é muito alto.

Se fosse rápido saber se o número é primo ou não, toda a segurança dos algoritmos de criptografia assimétrica estaria acabada.

Mas aqui lidaremos apenas com números pequenos. Nesses casos, um algoritmo possível seria: dado um número N , percorra todos os números entre 1 e N (inclusive) e conte o número de divisores* que N possui nesse intervalo. Se N possuir dois divisores, então N é primo, caso contrário não é.

Assim sendo, você poderia fazer um programa que dado o número N diga se ele é primo ou não?

Entrada

A entrada possui apenas um caso de teste. A única linha da entrada contém um inteiro N ($2 \leq N \leq 10^6$).

Saída

A saída deve ser “primo” (sem aspas) caso o número N seja primo, ou “nao primo” (sem aspas) caso contrário. **Atenção para a formatação da saída: a palavra “nao” não deve ter o acento na letra “a”, e não se esqueça de saltar uma linha no final.**

Exemplo de entrada	Exemplo de saída
50	nao primo

Exemplo de entrada	Exemplo de saída
2	primo

Exemplo de entrada	Exemplo de saída
1000000	nao primo

Exemplo de entrada	Exemplo de saída
7	primo

Exemplo de entrada	Exemplo de saída
999983	primo

*Observação: um número M é divisor de um número N , se o resto da divisão de M por N é zero. Em C, C++ ou Java o operador que retorna o resto da divisão é “%”. Nesse caso se $N \% M == 0$ então M é divisor de N .