

## Problema AJ

### Primos Arrojados

Limite de tempo: 1s

Autor: Arthur Luís Komatsu Aroeira

Kanela é um garoto fascinado por números primos (números que contêm exatamente 2 divisores positivos distintos). Desde que aprendeu na escola, sempre tentou encontrar primos em todos os números que via. Um de seus hobbies favoritos era escrever números primos em seu caderno. Um belo dia, Kanela resolveu apagar um número primo em seu caderno com uma borracha da direita para esquerda. Ao apagar o último dígito, Kanela notou que o número restante continuava sendo um primo. Ao apagar o último dígito deste, Kanela viu que o número restante também era primo. E assim por diante, conforme os dígitos eram apagados da direita para a esquerda, o número remanescente era primo até que fosse apagado completamente. Kanela achou isso o máximo e começou a chamar esses números de *primos arrojados*!

Por exemplo, o número 2393 é um primo arrojado, pois:

- 2393 é um número primo;
- retirando-se o último algarismo, 239 continua sendo um número primo;
- retirando-se o último algarismo, 23 também é um número primo; e
- removendo-se o último algarismo, tem-se 2 que é primo.

Ajude Kanela a dizer se um número é um *primo arrojado* ou não.

### Entrada

A primeira linha contém o inteiro  $T$  ( $1 \leq T \leq 2 \cdot 10^5$ ), que representa a quantidade de casos de teste.

Cada uma das próximas  $T$  linhas contém um número  $n$  ( $1 \leq n \leq 10^7$ ).

### Saída

Para cada caso de teste, imprima uma linha com ‘S’, caso o número  $n$  seja um *primo arrojado*, ou ‘N’, caso contrário.

### Exemplo

Entrada	Saída
11	N
1	S
2	S
7	N
10	N
11	N
22	S
23	N
123	N
173	S
233	S
2393	

### Notas

Observe que:

- 1 não é um primo arrojado, pois não é um número primo
- 2 é um primo arrojado, pois é primo e contém apenas um dígito
- 23 é um primo arrojado, pois 23 é primo e, retirando-se o último dígito, 2 também é primo
- 173 não é um primo arrojado, pois 1 não é primo, apesar de 17 e 173 serem primos
- 2393 é um primo arrojado, pois 2393, 239, 23 e 2 são todos números primos.

**Atenção:** É preciso usar um mecanismo de leitura e escrita eficiente para evitar o TLE na hora de escrever a saída do problema.

Para o C++, duas alternativas são:

1. Usar as funções `printf` / `scanf`.
2. Adicionar a linha `ios::sync_with_stdio(false)` no início da função `main`.

No caso de Java, é preciso fazer leitura com buffer (`Scan` resulta em TLE).