

# Dígitos Verificadores

Arquivo de entrada:	<code>entrada padrão</code>
Arquivo de saída:	<code>saída padrão</code>
Tempo limite:	1 segundo
Limite de memória:	256 megabytes

O primeiro erro de matemática registrado na história da humanidade provavelmente foi cometido por um contador 5 milênios atrás cujo nome é Kushim, que é o ser humano mais antigo cujo nome se tem registro. Junto com seu supervisor, Nisa, Kushim cuidava das contas de um armazém, contando a quantidade de cevada armazenada, assim como o registro das ordens de cerveja e receitas com as razões de cevada necessárias para prepará-las. É possível, portanto, que o primeiro erro da matemática seja só resultado de tentar fazer matemática enquanto se está bêbado.

Alguns milênios depois, por volta do ano 950, um astrônomo indiano, Aryabhata II, descreveu o uso de um método para verificar somas, subtrações, multiplicações e divisões numéricas. Esse método é chamado de “prova dos nove”. Ele consiste em pegar um número e tirar a sua soma digital (por exemplo, a soma digital de 2946 é  $2 + 9 + 4 + 6 = 21$ ), e em seguida pegar esse número módulo 9 (o módulo de 21, por exemplo, é 3).

O método da “prova dos nove” é quase equivalente a chamada “raiz digital”, que envolve pegar um número e repetir o processo de soma digital várias vezes (por exemplo, a soma digital de  $21 = 2 + 1 = 3$ ). Elas só não são equivalentes em relação aos múltiplos de nove, pois a soma digital de um múltiplo de nove não-zero é 9, enquanto que o método da “prova dos nove” iria resultar em 0. De toda forma, ambas são equivalentes  $(\text{mod } 9)$ .

É possível provar que a “raiz digital”  $(\text{mod } 9)$  e o método da “prova dos nove”  $(\text{mod } 9)$  são na verdade equivalentes ao próprio número  $(\text{mod } 9)$ . Isso permite que as propriedades dos números modulares seja extrapolada para esses outros métodos, o que permite que somas, subtrações, multiplicações e divisões sejam verificadas:

$$\begin{aligned}dr(a_1 + a_2) &= dr(dr(a) + dr(b)) \\ dr(ab) &= dr(dr(a)dr(b))\end{aligned}$$

Isso significa que se fizermos uma operação de soma ou de multiplicação, podemos fazer a raiz digital  $(\text{mod } 9)$  de forma separada nos operandos e na resposta, e o resultado tem que ser idêntico, temos então um dígito verificador.

Sabendo de tudo isso, vamos ajudar Nisa a achar os erros nas contas de Kushim usando o método dos “múltiplos de nove”. Primeiro, transcrevemos nossas tabelas de multiplicação (assim como nos tablets de Kushim) e números iniciais numa fita contínua para facilitar o reuso dos números e vamos verificar se o resultado da nossa soma ou multiplicação resulta no mesmo dígito verificador que é esperado. Vamos então ir colocando nos tablets os resultados das somas, até chegarmos na resposta final!

## Entrada

Na primeira linha, é dado o inteiro  $N$  ( $1 \leq N \leq 10^5$ ), o tamanho da fita de números. A linha que vem a seguir contém a fita de números que é uma sequência de dígitos.

A próxima linha contém o inteiro  $Q$  ( $1 \leq Q \leq 10^5$ ), o número de operações que vamos realizar. Em seguida, seguem  $Q$  linhas, cada uma com uma operação. No começo de cada linha há um identificador de operação inteiro  $O$ :

- Para  $O = 1$  ou  $O = 2$ , sendo que 1 representa soma e 2 representa multiplicação, temos dois inteiros  $A_1, B_1$  ( $1 \leq A_1 \leq B_1 \leq N$ ), dois inteiros  $A_2, B_2$  ( $1 \leq A_2 \leq B_2 \leq N$ ) e o dígito de verificação obtido  $D$  ( $0 \leq D \leq 8$ ). Os inteiros  $A_1$  e  $B_1$  representam o início e o fim inclusivo do primeiro operando na fita, enquanto que  $A_2$  e  $B_2$  representam o início e o fim inclusivo do segundo operando na fita.

- Para  $O = 3$ , temos um inteiro  $P$  ( $1 \leq P \leq N$ ) e um inteiro  $D$  ( $0 \leq D \leq 9$ ) que representam uma posição na fita e um dígito que será colocado naquela posição. Qualquer dígito previamente naquela posição é apagado.

## Saída

Para cada operação  $O = 1$  ou  $O = 2$ , imprima 'YES' se o dígito  $D$  informado bate com a raiz digital do resultado da operação  $\pmod{9}$ , ou 'NO' se não.

## Exemplo

entrada padrão
48
819235303009000007061060012102700594252702953000
15
2 1 1 6 6 4
2 2 3 7 8 3
2 4 4 9 11 6
2 5 5 12 14 0
3 15 4
3 17 5
1 15 16 17 19 7
1 20 22 23 25 4
1 26 29 30 33 4
2 6 6 34 34 7
2 35 35 7 8 0
2 36 36 12 14 0
1 37 38 39 41 7
3 46 6
1 42 44 45 48 4
saída padrão
YES
YES
YES
YES
YES
YES
YES
YES
YES
YES
YES
NO