

## C. Checksum

No mundo das redes de computadores, um problema crucial é garantir que os dados enviados de um computador para outro não sejam modificados no caminho. Para resolver isso, utiliza-se o *checksum*, uma técnica para verificar a integridade dos dados recebidos.

O *checksum* envolve a soma de dois valores incluídos na mensagem, que deve resultar em um valor binário onde todos os bits são 1. Por exemplo, se os números são 132 e 123, seus valores binários são 10000100 e 01111011, respectivamente. A soma desses valores binários é 11111111, confirmando a integridade dos dados.

Embora não seja infalível, essa técnica é útil e amplamente usada em datagramas UDP, um protocolo de rede comum.

O chefe de Bob pediu que ele escrevesse um programa que, dados dois números binários, verificasse se formam um *checksum* válido. Como Bob mentiu no currículo e não sabe programar, ele pediu sua ajuda para escrever o programa.

Para maior clareza, segue um caso de *checksum* **inválido**:

- representação binária do valor 130: 10000010
- representação binária do valor 124: 01111100
- $130 + 124 = 100000010 + 011111100 = 11111110$  (inválido)

### Entrada

A entrada consiste de dois números inteiros  $x_1, x_2$ : as representações binárias de 8 bits dos valores que fazem parte do *checksum*.

### Saída

Imprima "SIM" (sem as aspas) se os valores formarem um *checksum* válido. Imprima "NAO" (sem as aspas) caso contrário.

### Restrições

$x_1, x_2$  são valores **binários** no intervalo fechado  $[0, 11111111]$

Exemplo de entrada 01:	Exemplo de saída 01:
10110010 01001101	SIM

Exemplo de entrada 02:	Exemplo de saída 02:
11111101 00000001	NAO