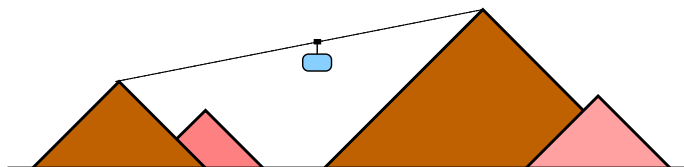


# Bondinho

Nome do arquivo: `bondinho.c`, `bondinho.cpp`, `bondinho.pas`, `bondinho.java`, `bondinho.js` ou `bondinho.py`

A turma do colégio vai fazer uma excursão na serra e todos os alunos e monitores vão tomar um bondinho para subir até o pico de uma montanha. A cabine do bondinho pode levar 50 pessoas no máximo, contando alunos e monitores, durante uma viagem até o pico. Neste problema, dado como entrada o número de alunos  $A$  e o número de monitores  $M$ , você deve escrever um programa que diga se é possível ou não levar todos os alunos e monitores em apenas uma viagem!



## Entrada

A primeira linha da entrada contém um inteiro  $A$ , representando a quantidade de alunos. A segunda linha da entrada contém um inteiro  $M$ , representando o número de monitores.

## Saída

Seu programa deve imprimir uma linha contendo o caractere **S** se é possível levar todos os alunos e monitores em apenas uma viagem, ou o caractere **N** caso não seja possível.

## Restrições

- $1 \leq A \leq 50$
- $1 \leq M \leq 50$

## Exemplos

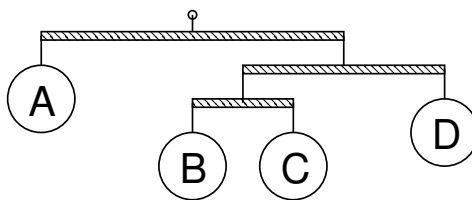
<b>Entrada</b> 10 20	<b>Saída</b> S
<b>Entrada</b> 12 39	<b>Saída</b> N
<b>Entrada</b> 49 1	<b>Saída</b> S

# MóBILE

Nome do arquivo: `mobile.c`, `mobile.cpp`, `mobile.pas`, `mobile.java`, `mobile.js` ou `mobile.py`

O móbile na sala da Maria é composto de três hastes exatamente como na figura abaixo. Para que ele esteja completamente equilibrado, com todas as hastes na horizontal, os pesos das quatro bolas  $A$ ,  $B$ ,  $C$  e  $D$  têm que satisfazer todas as seguintes três condições:

1.  $A = B + C + D$ ; e
2.  $B + C = D$ ; e
3.  $B = C$ .



Nesta tarefa, dados os pesos das quatro bolas, seu programa deve decidir se o móbile está ou não completamente equilibrado.

## Entrada

A entrada consiste de quatro linhas contendo, cada uma, um número inteiro, indicando os pesos das bolas. Os números são dados na ordem:  $A$ ,  $B$ ,  $C$  e  $D$ .

## Saída

Seu programa deve escrever uma única linha na saída, contendo o caractere “S” se o móbile estiver equilibrado, ou o caractere “N” se não estiver equilibrado.

## Restrições

- $1 \leq A, B, C, D \leq 1000$

## Exemplos

<b>Entrada</b> 12 3 3 6	<b>Saída</b> S
<b>Entrada</b> 2002 560 560 882	<b>Saída</b> N

# Prêmio do Milhão

*Nome do arquivo:* `premio.c`, `premio.cpp`, `premio.pas`, `premio.java`, `premio.js` ou `premio.py`

Alice e Bia criaram uma página na Internet com informações sobre o Macaco-prego-de-peito-amarelo, uma espécie em extinção. A página mostra como todos podem ajudar a manter o habitat natural para evitar que a espécie seja extinta.

Uma empresa gostou tanto da iniciativa de Alice e Bia que prometeu doar um prêmio para que as duas amigas possam realizar outras iniciativas semelhantes. A empresa decidiu que o prêmio seria dado quando a soma do número de acessos à página chegasse a 1 milhão.

Dada a lista de acessos diários que ocorreram à página de Alice e Bia, escreva um programa para determinar quantos dias foram necessários para a soma dos acessos chegar a 1 milhão e as amigas ganharem o prêmio.

## Entrada

A primeira linha da entrada contém um número inteiro  $N$ , que indica o número de dias que a lista contém. Cada uma das linhas seguintes contém um único inteiro  $A$ , o número de acessos em um dia. O primeiro número dado indica o número de acessos no primeiro dia, o segundo número dado indica o número de acessos no segundo dia, e assim por diante.

## Saída

Seu programa deve escrever na saída uma única linha, contendo um único número inteiro, o número de dias que foram necessários para a soma dos acessos à página de Alice e Bia chegar a 1000000.

## Restrições

- $1 \leq N \leq 10^3$ , ou seja, a lista tem no máximo 1000 números
- $0 < A \leq 10^6$ , ou seja, cada inteiro  $A$  da lista é positivo e menor do que ou igual a 1 milhão.
- A soma de todos os valores  $A$  da lista é maior do que ou igual a 1 milhão (ou seja, Alice e Bia certamente ganham o prêmio).

## Exemplos

Entrada	Saída
5 100 99900 400000 500000 600000	4
1 1000000	1

# Plantação de morango

*Nome do arquivo:* `morango.c`, `morango.cpp`, `morango.pas`, `morango.java`, `morango.js`,  
`morango.py2` ou `morango.py3`

Os administradores da Fazenda Fartura planejam criar uma nova plantação de morangos, no formato retangular. Eles têm vários locais possíveis para a nova plantação, com diferentes dimensões de comprimento e largura. Para os administradores, o melhor local é aquele que tem a maior área. Eles gostariam de ter um programa de computador que, dadas as dimensões de dois locais, determina o que tem maior área. Você pode ajudá-los?

## Entrada

A entrada contém quatro linhas, cada uma contendo um número inteiro. As duas primeiras linhas indicam as dimensões (comprimento e largura) de um dos possíveis locais. As duas últimas linhas indicam as dimensões (comprimento e largura) de um outro possível local para a plantação de morangos. As dimensões são dadas em metros.

## Saída

Seu programa deve escrever uma linha contendo um único inteiro, a área, em metros quadrados, do melhor local para a plantação, entre os dois locais dados na entrada.

## Restrições

- $1 \leq \text{comprimento} \leq 100$
- $1 \leq \text{largura} \leq 100$

## Exemplos

<b>Entrada</b> 30 8 11 56	<b>Saída</b> 616
<b>Entrada</b> 12 38 5 20	<b>Saída</b> 456

# Jogo de par ou ímpar

Nome do arquivo: `jogo.c`, `jogo.cpp`, `jogo.pas`, `jogo.java`, `jogo.js`, `jogo.py2` ou `jogo.py3`

Dois amigos, Alice e Bob, estão jogando um jogo muito simples, em que um deles grita ou “par” ou “ímpar” e o outro imediatamente responde ao contrário, respectivamente “ímpar” ou “par”. Em seguida, ambos exibem ao mesmo tempo uma mão cada um, em que alguns dedos estão estendidos e outros dobrados. Então eles contam o número total de dedos estendidos. Se a soma for par, quem gritou “par” ganha. Se a soma for ímpar, quem gritou “ímpar” ganha.

Por exemplo, suponhamos que a Alice gritou “par” e o Bob respondeu “ímpar”. Em seguida, Alice não deixou nenhum dos seus dedos estendidos, ao passo que Bob deixou três dedos estendidos. A soma então é três, que é ímpar, portanto Bob ganhou.

Seu programa deve determinar quem ganhou, tendo a informação de quem gritou par e o número de dedos estendidos de cada um.

## Entrada

A entrada contém três linhas, cada uma com um número inteiro,  $P$ ,  $D_1$  e  $D_2$ , nesta ordem. Se  $P = 0$  então Alice gritou “par”, ao passo que se  $P = 1$  então Bob gritou “par”. Os números  $D_1$  e  $D_2$  indicam, respectivamente, o número de dedos estendidos da Alice e do Bob.

## Saída

Seu programa deverá imprimir uma única linha, contendo um único número inteiro, que deve ser 0 se Alice foi a ganhadora, ou 1 se Bob foi o ganhador.

## Restrições

- $P = 0$  ou  $P = 1$
- $0 \leq D_1 \leq 5$
- $0 \leq D_2 \leq 5$

## Exemplos

<b>Entrada</b> 0 0 3	<b>Saída</b> 1
<b>Entrada</b> 1 0 3	<b>Saída</b> 0
<b>Entrada</b> 0 1 5	<b>Saída</b> 0

# Lâmpadas

Nome do arquivo: `lampadas.c`, `lampadas.cpp`, `lampadas.pas`, `lampadas.java`, `lampadas.js`,  
`lampadas.py2` ou `lampadas.py3`

Você está de volta em seu hotel na Tailândia depois de um dia de mergulhos. O seu quarto tem duas lâmpadas. Vamos chamá-las de  $A$  e  $B$ . No hotel há dois interruptores, que chamaremos de  $I_1$  e  $I_2$ . Ao apertar  $I_1$ , a lâmpada  $A$  troca de estado, ou seja, acende se estiver apagada e apaga se estiver acesa. Se apertar  $I_2$ , ambas as lâmpadas  $A$  e  $B$  trocam de estado.

As lâmpadas inicialmente estão ambas apagadas. Seu amigo resolveu bolar um desafio para você. Ele irá apertar os interruptores em uma certa sequência, e gostaria que você respondesse o estado final das lâmpadas  $A$  e  $B$ .

## Entrada

A primeira linha contém um número  $N$  que representa quantas vezes seu amigo irá apertar algum interruptor. Na linha seguinte seguirão  $N$  números, que pode ser 1, se o interruptor  $I_1$  foi apertado, ou 2, se o interruptor  $I_2$  foi apertado.

## Saída

Seu programa deve imprimir dois valores, em linhas separadas.

Na primeira linha, imprima 1 se a lâmpada  $A$  estiver acesa no final das operações e 0 caso contrário. Na segunda linha, imprima 1 se a lâmpada  $B$  estiver acesa no final das operações e 0 caso contrário.

## Restrições

- $1 \leq N \leq 10^5$

## Informações sobre a pontuação

- Em um conjunto de casos de teste equivalente a 20 pontos,  $N = 3$ .

## Exemplos

Entrada	Saída
3 1 2 2	1 0
Entrada	Saída
4 2 1 2 2	0 1