

# Kalkulator

Napisz program, który działa jak prosty kalkulator obsługujący pięć operacji: dodawanie, odejmowanie, mnożenie, dzielenie i obliczanie reszty z dzielenia liczb całkowitych.

## Wejście

Na wejście programu podana zostanie pewna nieokreślona liczba zestawów danych. Zestawy składają się z jednoznakowego symbolu operacji do wykonania (+ dodawanie, - odejmowanie, \* mnożenie, / dzielenie całkowitoliczbowe, % reszta z dzielenia) oraz następujących po nim dwóch liczb całkowitych. Poszczególne składowe zestawu zostaną rozdzielone spacjami, a same zestawy znakiem nowej linii. Liczba testów nie przekracza 100, wynik zawiera się w typie `int32`.

## Wyjście

Na wyjściu programu ma się pojawić ciąg liczb będących rezultatem wykonania pojawiających się na wejściu poleceń. Poszczególne liczby należy rozdzielić znakami nowej linii. **Uwaga!** Można założyć, że dane wejściowe nie zawierają polecenia dzielenia przez 0.

## Przykład

### Wejście:

```
+ 7 9
- 0 4
* 5 6
/ 8 3
% 5 2
```

### Wyjście:

```
16
-4
30
2
1
```

# Porównywanie dużych liczb

Napisz program, który sprawdza, czy pomiędzy daną parą liczb zachodzi podana relacja.

## Wejście

Na wejście programu podana zostanie pewna nieokreślona liczba zestawów danych. Zestawy składają się z liczby naturalnej, po której następuje spacja, symbol relacji ( $==$ ,  $!=$ ,  $>=$  lub  $<=$ ), spacja i druga liczba naturalna. Poszczególne zestawy zostaną rozdzielone znakiem nowej linii. Przyjmujemy, że żadna z liczb nie ma więcej niż 1000 cyfr.

## Wyjście

Na wyjściu programu ma się pojawić ciąg zerojedynekowy, którego  $i$ -ty element jest równy 1 lub 0 w zależności od tego, czy relacja zawarta w  $i$ -tym wczytanym z wejścia zestawie danych jest prawdziwa czy fałszywa. Poszczególne elementy tego ciągu należy rozdzielić znakami nowej linii.

## Przykład

### Wejście:

100 == 200

200 <= 100

200 >= 100

### Wyjście:

0

0

1

# Labirynt

Napisz program, który sprawdza, czy z danego labiryntu można wyjść.

## Wejście

Na wejście programu podana zostanie pewna nieokreślona ilość 100-znakowych łańcuchów, rozdzielonych znakiem nowej linii. Każdy z łańcuchów opisuje zawartość labiryntu. Labirynt ma kształt planszy o wymiarach 10 na 10; pole o współrzędnych  $(i,j)$  jest puste lub zajęte w zależności od tego, czy znakiem o numerze  $10*i+j$  jest O czy X. Na polu (0,0) znajduje się wejście do labiryntu; pole (9,9) zawiera wyjście.

## Wyjście

Na wyjściu programu ma się pojawić ciąg binarny, którego  $i$ -ty element jest równy 1 wtedy i tylko wtedy, gdy można wyjść z  $i$ -tego wczytanego z wejścia labiryntu. (Z labiryntu można wyjść, jeżeli znajdując się na początku w wejściu można przejść do wyjścia przechodząc wyłącznie przez puste pola i poruszając się wyłącznie pionowo lub poziomo.) Poszczególne elementy tego ciągu należy rozdzielić znakami nowej linii.

## Przykład

Wejście:

```
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000  
0000000000XXXXXXXXXX0000000000XXXXXXX00000000000000000000000000000000000000000000000000
```

**Wyjście:**

1  
0