

Zapałki

Limit pamięci: 64 MB

Bajtek jest leniwym perfekcjonistą. Właśnie przechodził swoją perfekcyjnie najkrótszą ścieżką z pracy do domu, kiedy zobaczył, że na chodniku ktoś ułożył działanie z zapałek. Oczywiście, było ono niepoprawne, co niezwykle zirytowało Bajtkę. Jako, że Bajtek jest leniwy postanowił, że jeżeli da się naprawić to działanie, przedstawiając dokładnie jedną zapałkę, to może poświęcić na to swój cenny czas.

Działanie ułożone z zapałek składa się z cyfr, znaków $+$, $-$ oraz dokładnie jednego znaku $=$. Bezpośrednio za każdym znakiem $+$ lub $-$ stoi cyfra. Znak $+$ nie stoi na początku działania, ani od razu po znaku $=$. Przenoszenie zapałki nie może zmieniać, usuwać ani tworzyć żadnych nowych znaków (można przenosić zapałkę tylko między cyframi). Nie można tworzyć nowych cyfr, ani usuwać istniejących (np zamieniając 4 na dwie 1). Na początku liczby w działaniu nie mają wiodących zer, ale po przełożeniu zapałki mogą mieć. Cyfry są na początku (oraz muszą być po przestawieniu zapałki) zapisane w następujący sposób:

0 1 2 3 4 5 6 7 8 9

Wejście

W jedynym wierszu wejścia znajduje się napis złożony z cyfr, oraz znaków $-$, $+$, $=$, zgodny z opisem w treści. Długość napisu nie przekroczy 10^6 . Długość żadnej liczby nie przekroczy 10 cyfr. Możesz założyć, że działanie nie jest poprawne.

Wyjście

Na wyjściu wypisz jeden napis, poprawione działanie w takim samym formacie co na wejściu, lub **no** jeżeli nie da się poprawić działania przenosząc jedną zapałkę. Jeżeli istnieje wiele poprawnych rozwiązań, twój program może wypisać dowolne z nich.

Przykłady

Wejście dla testu r1d0a:

1+1=3

Wyjście dla testu r1d0a:

1+1=2

Wejście dla testu r1d0b:

1-1=3+5

Wyjście dla testu r1d0b:

no

Wejście dla testu r1d0c:

-11-77=-34

Wyjście dla testu r1d0c:

-17-17=-34

Wejście dla testu r1d0d:

2023=2024

Wyjście dla testu r1d0d:

no

Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Punkty
4	Na wejściu nie będzie innych cyfr niż 4	4 s (C++) / 10 s (Python)	4
1	Na wejściu pojawią się tylko cyfry i znak =	3 s (C++) / 10 s (Python)	11
2	Na wejściu nie będzie znaków -	3 s (C++) / 10 s (Python)	15
3	Wszystkie liczby na wejściu będą jednocyfrowe	3 s (C++) / 10 s (Python)	15
5	Długość napisu $\leq 10^3$	3 s (C++) / 10 s (Python)	25
6	Brak dodatkowych ograniczeń	3 s (C++) / 10 s (Python)	30