Zadanie: CNF
CNF-SAT [B]

Potyczki Algorytmiczne 2016, runda 5. Dostępna pamięć: 256 MB. *25.11.2016*

Zbliża się kolejna Parada Równości $P = NP$. W tym roku nieformalny lider ruchu $P = NP$, Bajtazar, postanowił ostatecznie uciszyć swoich licznych przeciwników, ogłaszając na Paradzie dowód słynnej równości.

Dowód Bajtazara polega na pokazaniu algorytmu rozwiązującego znany NP-trudny problem *CNF-SAT* w czasie wielomianowym. W tym problemie dane jest n zmiennych logicznych $x_1, \ldots, x_n$ i formuła logiczna w tak zwanej *koniunkcyjnej postaci normalnej*. Taka formuła jest postaci

$$(l_{1,1} \vee \ldots \vee l_{1,q_1}) \wedge (l_{2,1} \vee \ldots \vee l_{2,q_2}) \wedge \ldots \wedge (l_{m,1} \vee \ldots \vee l_{m,q_m}),$$

gdzie każde z wyrażeń $(l_{i,1} \vee \ldots \vee l_{i,q_i})$ nazywamy *klauzulą*, a każde spośród wyrażeń $l_{i,j}$ jest *literałem*, czyli pewną zmienną lub zaprzeczeniem pewnej zmiennej spośród danych $x_1, \ldots, x_n$. Przyjmujemy, że żadna poprawna klauzula *nie zawiera* dwóch identycznych literałów. Dla $n = m = 3$, przykładową formułą w koniunkcyjnej postaci normalnej może być $(x_1 \vee \neg x_3) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_1 \vee x_2)$.

Problem CNF-SAT polega na rozstrzygnięciu, czy istnieje pewne wartościowanie zmiennych $x_1, \ldots, x_n$, dla którego dana formuła jest spełniona (to znaczy, jej wartością logiczną jest prawda).

Niestety, do dokończenia dowodu Bajtazarowi brakuje jednego kroku. Twierdzi on, że udało mu się sprowadzić* ogólny problem CNF-SAT do jego szczególnego przypadku, gdzie każda klauzula C danej formuły jest *spójna*, czyli ma następujące własności:

- Dla dowolnego i, x_i i $\neg x_i$ nie mogą być jednocześnie literałami C.
- Jeśli i, j, k są takie, że $i < j < k$ i zarówno zmienna x_i (lub jej zaprzeczenie) jak i x_k (lub $\neg x_k$) występują w klauzuli C, to x_j albo $\neg x_j$ także występuje w C.

Przykładowo, dla $n = 3$ klauzule (x_2) i $(\neg x_3 \vee \neg x_1 \vee x_2)$ są spójne, a klauzule $(x_2 \vee \neg x_2)$ i $(x_1 \vee \neg x_3)$ – nie.

Pomóż Bajtazarowi znaleźć efektywny algorytm rozwiązujący powyższy szczególny przypadek problemu CNF-SAT. Aby jeszcze bardziej go zadziwić, napisz program znajdujący liczbę wartościowań zmiennych $x_1, \ldots, x_n$ spełniających daną formułę CNF-SAT, składającą się z samych spójnych klauzul.

Wejście

W pierwszym wierszu wejścia znajduje się liczba całkowita n ($1 \leq n \leq 1\,000\,000$) oznaczająca liczbę zmiennych. W drugim wierszu wejścia mamy daną formułę CNF-SAT na zmiennych $x_1, \ldots, x_n$, składającą się z samych spójnych klauzul. Formuła dana jest w następującym formacie (patrz także przykład poniżej).

- Każda klauzula zaczyna się nawiasem otwierającym `(`, a kończy nawiasem zamykającym `)`.
- Literał x_i (dla $1 \leq i \leq n$) reprezentowany jest jako x*i*, a literał $\neg x_i$ jako ~x*i*, np. `x2` lub `~x15`.
- Sąsiednie literały w obrębie jednej klauzuli oddzielone są znakiem `v` (oznaczającym logiczne *lub*) otoczonym z obu stron pojedynczymi spacjami.
- Sąsiednie klauzule oddzielone są znakiem `^` (oznaczającym logiczne *i*) otoczonym z obu stron pojedynczymi spacjami.

Sumaryczna liczba literałów we wszystkich klauzulach danej formuły nie przekroczy $1\,000\,000$.

Wyjście

Na wyjście należy wypisać liczbę wartościowań zmiennych $x_1, \ldots, x_n$ spełniających daną na wejściu formułę, modulo $10^9 + 7$.

Przykład

Dla danych wejściowych:

```
3
(x2) ^ (x3 v ~x2) ^ (x2 v x1 v ~x3)
```

poprawnym wynikiem jest:

```
2
```

Wyjaśnienie do przykładu: Dana formuła jest spełniona tylko dla dwóch wartościowań: $(0, 1, 1)$ i $(1, 1, 1)$.

*Bajtazar zapomniał wspomnieć, czy jego redukcja działała w czasie wielomianowym...