

Opis zadania

Twoim zadaniem jest zaimplementowanie programu, który będzie weryfikował, czy podana formuła rachunku zdań jest tautologią rachunku zdań. Program ponadto będzie obsługiwał różne zapytania dotyczące formuły (szczegóły poniżej).

W programie zabronione jest używanie pojemników z biblioteki standardowej. Zabronione jest także wykorzystanie plików nagłówkowych innych niż `cstring`, `cmath`, `cstdio` i `iostream`.

Pisząc program możesz założyć, że wszystkie dane wejściowe są poprawne, tzn. tryby i wyrażenia są podane zgodnie z opisanymi poniżej regułami.

Format danych wejściowych

Wejście do programu będzie składało się z pewnej liczby linii. W każdej linii będą się znajdowały następujące dane:

tryb parametry trybu wyrażenie

Tryb jest jednoliterową komendą, która określa rodzaj zadania do wykonania przez program. Możliwe tryby: T, P, F, 0, 1, W (opis poszczególnych trybów znajduje się poniżej).

Parametry trybu to dodatkowy argument dla wybranych trybów, określający sposób działania tych trybów. Szczegółowy opis znajduje się przy opisie trybów.

Wyrażenie to formuła rachunku zdań, która będzie przedmiotem obliczeń programu w zadanym trybie. Szczegółowy opis konstrukcji wyrażeń znajduje się poniżej.

Wyjątkiem od powyższej reguły dotyczącej poszczególnych linii wejściowych będzie ostatnia linia danych, która będzie zawierała tylko literę W oznaczającą (W)yjście z programu.

Reguły dotyczące wyrażeń

W każdym wyrażeniu (formule) znajdują się zmienne zdaniowe. Zmienne zdaniowe będą reprezentowane tylko przez małe litery alfabetu angielskiego (a - z). Zatem formuły nie będą zawierały więcej niż 26 różnych zmiennych zdaniowych.

W wyrażeniu może pojawić się wielokrotnie jednoargumentowy funktor zdaniotwórczy - negacja. W zapisie formuły będzie ona reprezentowana przez znak tyldy (znak ~).

W formule mogą pojawić się jeszcze funktory dwuargumentowe:

- alternatywa (znak |),
- koniunkcja (znak &),
- implikacja (znak >),
- równoważność (znak =).

Dodatkowo, w wyrażeniu mogą pojawić się okrągłe nawiasy: (i). Żadne inne znaki nie mogą pojawić się w wyrażeniu pojawiającym się na wejściu.

Stosowanie wyżej wymienionych symboli podlega ściśle określonym regułom.

Każde wyrażenie, które jest zdaniem składającym się z funktora dwuargumentowego, musi zostać ujęte w nawiasy, np: $(p=p)$, $(p>(q>p))$, $((p\&q)=(q\&p))$. Wyrażenie nie będzie zawierało redundantnych nawiasów, czyli np. $((p=p))$ albo $(p>(q>(p)))$.

Symbol negacji może pojawić się bezpośrednio przed zmienną zdaniową, np. $(p|\sim p)$, $((p>q)=(\sim p|q))$. Ponadto, symbol negacji może pojawić się przed wyrażeniem ujętym w nawiasy, np. $(\sim(p>q)=(p\&\sim q))$, $\sim(p\&\sim p)$. Wielokrotne zaprzeczanie zmiennej zdaniowej lub wyrażenia będzie oznaczane przez wielokrotne użycie znaku \sim , bez dodatkowego ujmowania w nawiasy kolejnych zaprzeczanych zmiennych lub wyrażeń, np. $(\sim\sim p>p)$, $\sim\sim\sim(p\&\sim p)$.

Wszystkie zmienne i symbole pojawiające się w wyrażeniu występują bezpośrednio po sobie, tzn. nie są rozdzielane żadnymi białymi znakami. Zakładamy, że wyrażenie nie będzie miało więcej niż 1000 symboli zmiennych i zdaniotwórczych wymienionych wyżej łącznie.

Tryby działania programu

Program będzie obsługiwał kilka trybów:

Tryb **Tautologia** (oznaczany jednoliterową komendą **T**). W tym trybie program weryfikuje, czy podane po komendzie T wyrażenie jest tautologią. Dla tego trybu nie przewidziano parametrów. Program wypisze na standardowym wyjściu słowo TAK, jeśli podane wyrażenie jest tautologią albo słowo NIE, jeśli wyrażenie nie jest tautologią. Po wydrukowaniu wyniku zostanie wstawiony znak nowej linii.

Tryb **Ilość Prawd** (oznaczany jednoliterową komendą **P**). W tym trybie program zlicza, ile wartościowań dla podanego wyrażenia jest prawdziwych. Dla tego trybu nie przewidziano parametrów. Program wypisze na standardowym wyjściu liczbę wartościowań, dla których wyrażenie jest prawdą. Po wypisaniu wyniku zostanie wstawiony znak nowej linii.

Tryb **Ilość Fałszów** (oznaczany jednoliterową komendą **F**). W tym trybie program zlicza, ile wartościowań dla podanego wyrażenia jest fałszywych. Dla tego trybu nie przewidziano parametrów. Program wypisze na standardowym wyjściu liczbę wartościowań, dla których wyrażenie jest fałszywe. Po wypisaniu wyniku zostanie wstawiony znak nowej linii.

Tryb **Podstawienie 1** (oznaczany jednoliterową komendą **1**). W tym trybie program przyjmuje dodatkowy parametr trybu, który jest małą literą i reprezentuje zmienną zdaniową występującą w wyrażeniu. Program podstawia wartość 1 (prawdę) za zmienną zdaniową wskazaną w parametrze trybu w każdym miejscu jej wystąpienia w danym wyrażeniu, i zweryfikuje, czy takie wyrażenie z zastosowanym podstawieniem jest tautologią. Jeśli jest, program wypisze na standardowym wyjściu słowo TAK i wydrukuje znak końca linii, jeśli nie jest, wypisze słowo NIE i również wydrukuje znak końca linii.

Tryb **Podstawienie 0** (oznaczany jednoliterową komendą **0**). W tym trybie program przyjmuje dodatkowy parametr trybu, który jest małą literą i reprezentuje zmienną zdaniową występującą w wyrażeniu. Program podstawia wartość 0 (fałsz) za zmienną zdaniową wskazaną w parametrze trybu w każdym miejscu jej wystąpienia w danym wyrażeniu, i zweryfikuje, czy takie wyrażenie z zastosowanym podstawieniem jest tautologią. Jeśli jest, program wypisze na standardowym wyjściu słowo TAK i wydrukuje znak końca linii, jeśli nie jest, wypisze słowo NIE i również wydrukuje znak końca linii.

Program zakończy swoje działanie w momencie, kiedy na wejściu przeczyta komendę **W** (oznaczającą wyjście z programu).

Test jawny

Wejście	Wyjście
T (p=p)	TAK
T (p> (q>p))	TAK
T ((p&q) = (q&p))	TAK
P (p ~p)	2
F ((p>q) = (~p q))	0
T (~ (p>q) = (p&~q))	TAK
T (p&~p)	NIE
P ((~~p>p) &q)	2
F ((~~p>p) &q)	2
F ~~~ (p&~p)	0
0 p (p>q)	TAK
0 q (p>q)	NIE
1 p (p>q)	NIE
1 q (p>q)	TAK
W	