

Задача А. Разбор утверждения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На вход вашей программе дается утверждение в следующей грамматике:

$$\begin{aligned}\langle \text{Файл} \rangle &::= \langle \text{Выражение} \rangle \\ \langle \text{Выражение} \rangle &::= \langle \text{Дизъюнкция} \rangle \mid \langle \text{Дизъюнкция} \rangle \text{ '}' \langle \text{Выражение} \rangle \\ \langle \text{Дизъюнкция} \rangle &::= \langle \text{Конъюнкция} \rangle \mid \langle \text{Дизъюнкция} \rangle \text{ '}' \langle \text{Конъюнкция} \rangle \\ \langle \text{Конъюнкция} \rangle &::= \langle \text{Отрицание} \rangle \mid \langle \text{Конъюнкция} \rangle \text{ '}' \langle \text{Отрицание} \rangle \\ \langle \text{Отрицание} \rangle &::= \text{ '!' } \langle \text{Отрицание} \rangle \mid \langle \text{Переменная} \rangle \mid \text{ '(' } \langle \text{Выражение} \rangle \text{ ')' } \\ \langle \text{Переменная} \rangle &::= (\text{ 'A' } \dots \text{ 'Z' }) \{ \text{ 'A' } \dots \text{ 'Z' } \mid \text{ '0' } \dots \text{ '9' } \mid \text{ '}' \}^*\end{aligned}$$

Имена переменных не содержат пробелов. Между символами оператора '}' нет пробелов. В остальных местах пробелы могут присутствовать. Символы табуляции и возврата каретки должны трактоваться как пробелы.

Вам требуется написать программу, разбирающую утверждение и строящую его дерево разбора, и выводящую полученное дерево в единственной строке без пробелов в следующей грамматике:

$$\begin{aligned}\langle \text{Файл} \rangle &::= \langle \text{Вершина} \rangle \\ \langle \text{Вершина} \rangle &::= \text{ '(' } \langle \text{Знак} \rangle \text{ '}' \langle \text{Вершина} \rangle \text{ '}' \langle \text{Вершина} \rangle \text{ ')' } \\ &\quad \mid \text{ '(' } \langle \text{Вершина} \rangle \text{ ')' } \\ &\quad \mid \langle \text{Переменная} \rangle \\ \langle \text{Знак} \rangle &::= \text{ '}' \mid \text{ '}' \mid \text{ '}' \\ \langle \text{Переменная} \rangle &::= (\text{ 'A' } \dots \text{ 'Z' }) \{ \text{ 'A' } \dots \text{ 'Z' } \mid \text{ '0' } \dots \text{ '9' } \mid \text{ '}' \}^*\end{aligned}$$

Формат входных данных

В единственной строке входного файла дано утверждение в грамматике из условия. Размер входного файла не превышает 100 КБ.

Формат выходных данных

В единственной строке выходного файла выведите дерево разбора утверждения без пробелов.

Примеры

стандартный ввод
!A&!B->!(A B)
стандартный вывод
(->,(& , (!A) , (!B)) , (! (, A , B)))
стандартный ввод
P1' ->!QQ->!R10&S !T&U&V
стандартный вывод
(-> , P1' , (-> , (!QQ) , (, (& , (!R10) , S) , (& , (& , (!T) , U) , V)))

Задача В. Минимизация доказательства

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	512 мегабайт

На вход вашей программе дается доказательство утверждения в следующей грамматике:

```

    <Файл> ::= <Контекст> ' |- ' <Выражение> '\n' <Строка>*
    <Контекст> ::= <Выражение> [ ' , ' <Выражение> ]*
    <Строка> ::= <Выражение> '\n'
    <Выражение> ::= <Выражение> '&' <Выражение>
    | <Выражение> '!' <Выражение>
    | <Выражение> '->' <Выражение>
    | '!' <Выражение>
    | '(' <Выражение> ')'
    | <Переменная>
    <Переменная> ::= ('A' ... 'Z') { 'A' ... 'Z' | '0' ... '9' | ',' }*
```

Операторы '&' и '!' левоассоциативны. Оператор '->' правоассоциативен. Операторы в порядке уменьшения приоритета: '!', '&', '|', '->'.

Имена переменных не содержат пробелов. Между символами одного оператора нет пробелов ('->' и '|-'). В остальных местах пробелы могут присутствовать. Символы табуляции и возврата каретки должны трактоваться как пробелы.

Требуется проверить доказательство на корректность. Если оно неверно, выведите «Proof is incorrect». Иначе минимизируйте и проаннотируйте доказательство.

Под минимизацией доказательства подразумевается создание нового доказательства такого, что:

- Новое доказательство доказывает то же самое утверждение в том же самом контексте
- Строки нового доказательства являются подпоследовательностью строк исходного доказательства
- В новом доказательстве ни одно выражение не встречается в нескольких строках
- В новом доказательстве нет неиспользуемых выражений, т.е. все выражения, кроме последнего, должны использоваться одним или более применением правила Modus Ponens.

Под аннотированием доказательства подразумевается:

- Все строки должны быть пронумерованы
- Каждая строка должна содержать пояснение, как она была выведена:
 1. Аксиома: номер аксиомы
 2. Предположение: номер предположения
 3. Modus Ponens: номера строк, в которых записаны выражения, используемые для вывода выражения в текущей строке

Формат входных данных

Во входном файле задано доказательство в приведенной выше грамматике. Размер входного файла не превышает 10 МБ.

Формат выходных данных

Если данное доказательство является некорректным, в единственной строке выходного файла должна быть запись «Proof is incorrect».

Иначе в файле должно быть минимизированное проаннотированное корректное доказательство. Каждая строка, кроме последней, должна быть использована хотя бы в одной аннотации Modus Ponens. Подробный формат аннотаций смотрите в примерах.

Примеры

стандартный ввод
<pre> - A -> A A & A -> A A -> A -> A A -> (A -> A) -> A A & A -> A (A -> A -> A) -> (A -> (A -> A) -> A) -> A -> A (A -> (A -> A) -> A) -> A -> A A & A -> A A -> A</pre>
стандартный вывод
<pre> - (A -> A) [1. Ax. sch. 1] (A -> (A -> A)) [2. Ax. sch. 1] (A -> ((A -> A) -> A)) [3. Ax. sch. 2] ((A -> (A -> A)) -> ((A -> ((A -> A) -> A)) -> (A -> A))) [4. M.P. 3, 1] ((A -> ((A -> A) -> A)) -> (A -> A)) [5. M.P. 4, 2] (A -> A)</pre>
стандартный ввод
<pre>A->B, !B - !A A->B !B !B -> A -> !B A -> !B (A -> B) -> (A -> !B) -> !A (A -> !B) -> !A !A</pre>
стандартный вывод
<pre>(A -> B), !B - !A [1. Hypothesis 1] (A -> B) [2. Hypothesis 2] !B [3. Ax. sch. 1] (!B -> (A -> !B)) [4. M.P. 3, 2] (A -> !B) [5. Ax. sch. 9] ((A -> B) -> ((A -> !B) -> !A)) [6. M.P. 5, 1] ((A -> !B) -> !A) [7. M.P. 6, 4] !A</pre>
стандартный ввод
<pre>A, C - B' B'</pre>
стандартный вывод
<pre>Proof is incorrect</pre>

Задача С. Теорема Гливенко

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 15 секунд
Ограничение по памяти: 512 мегабайт

На вход вашей программе дается **корректное** доказательство утверждения α в классическом исчислении высказываний. Доказательство записано с использованием грамматики из предыдущего задания.

Вам требуется построить корректное доказательство утверждения $\neg\neg\alpha$ в интуиционистском исчислении высказываний.

Формат входных данных

Во входном файле задано доказательство утверждения α в классическом исчислении высказываний. Размер входного файла не превышает 5 КБ.

Формат выходных данных

Файл должен содержать корректное доказательство утверждения $\neg\neg\alpha$ в интуиционистском исчислении высказываний в том же контексте, что доказательство α во входном файле.

Пример

стандартный ввод
A - A A
стандартный вывод
A - !!A A (A -> (!A -> A)) (!A -> A) (!A -> (!A -> !A)) ((!A -> (!A -> !A)) -> ((!A -> (!A -> !A) -> !A)) -> (!A -> !A))) ((!A -> ((!A -> !A) -> !A)) -> (!A -> !A)) (!A -> ((!A -> !A) -> !A)) (!A -> !A) ((!A -> A) -> ((!A -> !A) -> !!A)) ((!A -> !A) -> !!A) !!A

Замечание

В классическом исчислении высказываний используются следующие схемы аксиом:

- (1) $\alpha \rightarrow \beta \rightarrow \alpha$
- (2) $(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma)$
- (3) $\alpha \rightarrow \beta \rightarrow \alpha \& \beta$
- (4) $\alpha \& \beta \rightarrow \alpha$
- (5) $\alpha \& \beta \rightarrow \beta$
- (6) $\alpha \rightarrow \alpha \vee \beta$
- (7) $\beta \rightarrow \alpha \vee \beta$
- (8) $(\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow (\alpha \vee \beta \rightarrow \gamma)$
- (9) $(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha$
- (10) $\neg\neg\alpha \rightarrow \alpha$

В интуиционистском исчислении высказываний 10-я схема аксиом заменяется на:

- (10) $\alpha \rightarrow \neg\alpha \rightarrow \beta$