

## 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. Перестроение доказательства

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В данной задаче требуется проверить доказательство выражения в гильбертовском варианте интуиционистского исчисления высказываний и перестроить его в доказательство в натуральном выводе.

### Формат входных данных

На вход дается доказательство утверждения в соответствии со следующей грамматикой:

```

    <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.

### Формат выходных данных

Если входное доказательство неверно, выведите «Proof is incorrect at line  $n$ », где  $n$  — номер первой недоказанной строки в доказательстве.

Иначе выведите доказательство. Каждая строка доказательства — узел дерева, пустых строк быть не должно (кроме последней строки). Дочерние узлы указываются перед родительским узлом. В начале строки — уровень узла в квадратных скобках, потом через пробел — формула, в конце строки — обозначение правила, также через пробел и в квадратных скобках. Для обозначения лжи используйте комбинацию «\_!\_»: подчёркивание (ASCII 95), вертикальная черта (ASCII 124), подчёркивание (ASCII 95). В остальном следуйте формату из примеров.

Доказанное во входном файле высказывание должно быть заключением самого верхнего правила. В данном высказывании отрицание термов ( $\neg\varphi$ ) передавайте как  $(\varphi \rightarrow \perp)$ . В доказательстве вы можете пользоваться следующими правилами. Посылки правил должны идти в указанном порядке, переставлять их нельзя — однако, гипотезы в контексте могут быть произвольно переставлены.

Обозначение	Посылки	Заключение
$Ax$		$\Gamma, \varphi \vdash \varphi$
$E \rightarrow$	$\Gamma \vdash \varphi \rightarrow \psi, \Gamma \vdash \varphi$	$\Gamma \vdash \psi$
$I \rightarrow$	$\Gamma, \varphi \vdash \psi$	$\Gamma \vdash \varphi \rightarrow \psi$
$I \&$	$\Gamma \vdash \varphi, \Gamma \vdash \psi$	$\Gamma \vdash \varphi \& \psi$
$E1 \&$	$\Gamma \vdash \varphi \& \psi$	$\Gamma \vdash \varphi$
$E2 \&$	$\Gamma \vdash \varphi \& \psi$	$\Gamma \vdash \psi$
$I \vee$	$\Gamma \vdash \varphi$	$\Gamma \vdash \varphi \vee \psi$
$I \vee$	$\Gamma \vdash \psi$	$\Gamma \vdash \varphi \vee \psi$
$E \vee$	$\Gamma, \varphi \vdash \rho, \Gamma, \psi \vdash \rho, \Gamma \vdash \varphi \vee \psi$	$\Gamma \vdash \rho$
$E \perp$	$\Gamma \vdash \perp$	$\Gamma \vdash \varphi$

## Примеры

стандартный ввод	стандартный вывод
$A \mid - A \rightarrow A$ $A \rightarrow A \rightarrow A$ $A$ $A \rightarrow A$	$[3] A, A, A \mid - A [Ax]$ $[2] A, A \mid - A \rightarrow A [I \rightarrow]$ $[1] A \mid - A \rightarrow A \rightarrow A [I \rightarrow]$ $[1] A \mid - A [Ax]$ $[0] A \mid - A \rightarrow A [E \rightarrow]$
$A, C \mid - B'$ $B'$	Proof is incorrect at line 2

## Замечание

Рассмотрим доказательство  $A \rightarrow A$  (гильбертовский стиль). Входной файл, соответствующий доказательству, мог бы быть таким:

```

|-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

```

Поскольку утверждение может быть доказано следующим натуральным выводом:

$$\frac{A \vdash A}{\vdash A \rightarrow A}$$

То, соответственно, текст ниже будет корректным ответом на задачу.

```

[1] A|-A [Ax]
[0] |-A->A [I->]

```

## Задача С. Формальная арифметика 2021

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Напишите программу, проверяющую доказательство в формальной арифметике на корректность.

### Формат входных данных

```
⟨Файл⟩ ::= ⟨заголовок⟩'⟨n⟩⟨доказательство⟩
⟨заголовок⟩ ::= 'I-'⟨выражение⟩
⟨доказательство⟩ ::= {⟨выражение⟩'⟨n⟩}+
⟨выражение⟩ ::= ⟨дизъюнкция⟩ | ⟨дизъюнкция⟩'-'⟨выражение⟩
⟨дизъюнкция⟩ ::= ⟨конъюнкция⟩ | ⟨дизъюнкция⟩'I'⟨конъюнкция⟩
⟨конъюнкция⟩ ::= ⟨унарное⟩ | ⟨конъюнкция⟩'&'⟨унарное⟩
⟨унарное⟩ ::= ⟨предикат⟩ | '!'⟨унарное⟩ | '('⟨выражение⟩')'
              | ('@'|'?')⟨переменная⟩.⟨выражение⟩
⟨переменная⟩ ::= 'a'... 'z'
⟨предикат⟩ ::= 'A'... 'Z'
              | ⟨терм⟩'='⟨терм⟩
⟨терм⟩ ::= ⟨слагаемое⟩ | ⟨терм⟩'+'⟨слагаемое⟩
⟨слагаемое⟩ ::= ⟨умножаемое⟩ | ⟨слагаемое⟩'*'⟨умножаемое⟩
⟨умножаемое⟩ ::= ⟨переменная⟩ | '('⟨терм⟩')'
              | '0' | ⟨умножаемое⟩','
```

Коды символов: символ апострофа (') — 0x27, вертикальная черта (|) — 0x7c.

### Формат выходных данных

Если доказательство корректно, проаннотируйте его. Первая строка должна повторять строку из входного файла, остальные строки доказательства должны быть предварены аннотацией:

- [n. Ax. sch. k], где  $n$  — номер выражения, а  $k$  — номер схемы аксиом: либо число от 1 до 12, либо A9.
- [n. Ax. k], где  $k$  — значение от A1 до A8.
- [n. M.P. k, 1], [n. ?-intro k], [n. @-intro k] — для правил вывода. Смысл индексов для M.P.: если доказательство представлено формулами  $\delta_i$ , то запись слева означает  $\delta_l \equiv \delta_k \rightarrow \delta_n$ .

Аннотации перечислены в порядке предпочтения: если выражение может быть обосновано, допустим, как аксиома A8 или как M.P., в ответе должно быть указано Ax. A8. В случае пересечения аксиом/схем указывайте аксиому/схему с минимальным номером; арифметические аксиомы/схемы идут после логических. Если выражение может быть получено при помощи одного правила вывода несколькими способами, предпочтение должно отдаваться наиболее ранним ссылкам в лексикографическом порядке: M.P. 1,10 предпочтительнее M.P. 10,1. Modus Ponens предпочтительнее правил с кванторами, правило с квантором существования предпочтительнее правила с квантором всеобщности (даже если номер исходной формулы для правила с квантором существования меньше). Также, аксиомы предпочтительнее правил вывода.

В выражениях должны быть расставлены все скобки в точности по одному разу (т.е. скобки вокруг всех унарных и бинарных выражений — кроме апострофов).

Если доказательство некорректно, выведите одну из следующих строк, в зависимости от типа ошибки. Ваша программа должна находить первое некорректное выражение в доказательстве, и для него указывать тип ошибки с минимальным номером (в соответствии со списком ниже):

1. Expression n: variable v occurs free in ?-rule.
2. Expression n: variable v occurs free in @-rule.
3. Expression n: variable v is not free for term t in ?-axiom.
4. Expression n: variable v is not free for term t in @-axiom.
5. Expression n is not proved.
6. The proof proves different expression.

Все строки доказательства, предшествующие некорректной, должны быть проаннотированы.

Столь подробные правила введены для того, чтобы упростить проверяющую программу: ответ сравнивается с эталонным на равенство; будьте внимательны.

## Пример

стандартный ввод
$\neg a + 0 = a$ $((a) + 0) = a$ $(\exists y. y + 0 * 0' = y) \rightarrow (\exists x. \exists y. x = y)$
стандартный вывод
$\neg ((a + 0) = a)$ [1. Ax. A5] $((a + 0) = a)$ Expression 2: variable x is not free for term $(y + (0 * 0'))$ in ?-axiom.