

## Задача 1. Функция MyPrintf

Источник: базовая  
Имя входного файла:  
Имя выходного файла:  
Ограничение по времени: 1 секунда  
Ограничение по памяти: разумное

Реализуйте свой собственный аналог функции `printf`. Формального списка того, какие спецификаторы должны быть реализованы нет, но не менее половины следующего списка должна работать:

- `%d` для вывода целых чисел,
- `%llu` для длинных и беззнаковых чисел,
- `%10d` для вывода целых чисел с выравниванием,
- `/*d` для вывода целых чисел с выравниванием, задающимся через аргумент,
- `%s` для строк,
- `%c` для отдельных символов,
- `%lf` для вещественных чисел,
- `%.5f` для форматирования при выводе вещественных чисел,
- `%p` для вывода адресов,
- `%n`, если станет скучно.

Так же обязательным является поддержка экранирования: `%%` в форматной строке должно превращаться в один символ `%` при выводе.

Оценка за задачу определяется временем её сдачи, начиная с 25 баллов в первые две недели со дня выдачи задания, затем она будет понижаться по 5 баллов в неделю до 5 баллов.

Поскольку основной целью данной задачи является разбор переменного количества аргументов, а не преобразования двоичного вида чисел в десятичный, то для преобразования чисел в строку разрешается использовать функцию `sprintf`.

Для всего остального запрещается использовать функции из семейства `printf`.

Про функцию `printf` можно почитать: <http://en.cppreference.com/w/c/io/fprintf>

## Задача 2. Сравнение асимптотик

Источник:	основная
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	разумное

В курсе всё больше и больше делается акцент на улучшение асимптотического времени работы различных операций. При этом важно понимать, при каких изменениях асимптотическое время работы становится лучше (т.е. быстрее), а при каких хуже. В данной задаче предлагается реализовать сравнение для наиболее часто встречающихся асимптотик.

В данной задаче асимптотическое время работы задаётся как функция вида:

$$T(N) = O(p^N \cdot N^s \cdot \log^l N)$$

Здесь  $p \geq 1$ ,  $s \geq 0$  и  $l \geq 0$  — произвольные вещественные числа. Легко видеть, что в этот класс попадают, например, асимптотики сортировки слиянием  $O(N \log N)$ , бинарного поиска  $O(\log N)$ , перебора всех  $N$ -битных чисел  $O(2^N N)$ .

### Формат входных данных

В первой строке входного файла записано число  $Q$  — сколько тестовых случаев нужно обработать ( $1 \leq Q \leq 10^5$ ). Далее идёт  $2Q$  строк, каждая пара строк описывает один тестовый случай, то есть две асимптотики, которые надо сравнить.

Асимптотика в полном виде записывается как: `"O( p^N N^s logN^l )"` (без кавычек). В полном виде в ней три части, обязательно отделённые друг от друга и от окружающих скобок пробелом. Других пробелов нет. Части могут быть записаны в произвольном порядке.

Вместо букв  $p$ ,  $s$  и  $l$  в описании асимптотики записаны вещественные числа, задающие соответствующие коэффициенты. Все вещественные числа записаны с не более чем тремя знаками после десятичной точки, и лежат в пределах от 0 до 10 включительно. Кроме того, для коэффициента  $p$  верно:  $p \geq 1$ .

Кроме того, некоторые части могут быть опущены: в таком случае в произведении этой части нет. Если опущены все три части, то асимптотика будет записана в виде `"O( 1 )"` (без кавычек). Наконец, в компонентах  $N^s$  и  $\log N^l$  может быть опущена степень: в таком случае она равна единице. Если степень опущена, то в описании отсутствует как вещественное число  $s$  или  $l$ , так и символ крышки непосредственно до него.

**Замечание:** рекомендуется использовать `gets`, `strtok`, `sscanf` и прочие стандартные функции для чтения асимптотики.

### Формат выходных данных

В выходных данных должно быть ровно  $Q$  целых чисел, по одному числу в строке. Если в запросе первая асимптотика меньше второй, число должно быть равно  $-1$ . Если первая асимптотика больше второй, то нужно вывести 1. Наконец, если они совпадают, то нужно вывести 0.

## Пример

input.txt	output.txt
6	-1
$O(2^N N^{3.5} \log N^{7.3})$	0
$O(2^N N^4 \log N^{7.267})$	-1
$O(N^{3.5} \log N^7)$	1
$O(\log N^{7.000} N^{3.5})$	1
$O(1)$	-1
$O(N^2)$	
$O(N^{0.5})$	
$O(\log N^7)$	
$O(2^N N)$	
$O(2^N)$	
$O(N \log N)$	
$O(N^{1.5})$	