Задача A. X+Y

Имя входного файла:	`xy.in`
Имя выходного файла:	`xy.out`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

Вам заданы x и y, выведите $x + y$!

Формат входного файла

Заданы x и y ($1 \le x, y \le 1000$).

Формат выходного файла

Выведите $x + y$.

Примеры

xy.in	xy.out
20 10	30
10 20	30

Задача B. Скобки

Имя входного файла:	`brackets.in`
Имя выходного файла:	`brackets.out`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

Требуется определить, является ли правильной данная последовательность круглых, квадратных и фигурных скобок.

Формат входного файла

В единственной строке входного файла записано подряд N скобок ($1 \le N \le 10^5$).

Формат выходного файла

В выходной файл вывести «`YES`», если данная последовательность является правильной, и «`NO`» в противном случае.

Примеры

brackets.in	brackets.out
()	YES
(((	NO

Задача C. Парикмахерская

Имя входного файла:	`saloon.in`
Имя выходного файла:	`saloon.out`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

В парикмахерской работает один мастер. Он тратит на одного клиента ровно 20 минут, а затем сразу переходит к следующему, если в очереди кто-то есть, либо ожидает, когда придет следующий клиент.

Даны времена прихода клиентов в парикмахерскую (в том порядке, в котором они приходили).

Так же у каждого клиента есть храктеристика, назоваемая *степенью нетерпения*. Она показывает сколько человек может максимально находится в очереди перед клиентом, чтобы он дождался своей очереди и не ушел раньше. Если в момент прихода клиента в очереди находится больше людей, чем степень его нетерпения, то он решает не ждать своей очереди и уходит. Клиент, который обслуживается в данный момент так же считается находящимся в очереди.

Требуется для каждого клиента указать время его выхода из парикмахерской.

Формат входного файла

В первой строке вводится натуральное число N, не превышающее 100 — количество клиентов.

В следующих N строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 23, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 100) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

Гарантируется, что всех клиентов успеют обслужить до полуночи.

Если для каких-то клиентов время окончания обслуживания одного клиента и время прихода другого совпадают, то можно считать, что в начале заканчивается обслуживание первого клиента, а потом приходит второй клиент.

Формат выходного файла

В выходной файл выведите N пар чисел: времена выхода из парикмахерской 1-го, 2-го, ..., N-го клиента (часы и минуты). Если на момент прихода клиента человек в очереди больше, чем степень его нетерпения, то можно считать, что время его ухода равно времени прихода.

Примеры

saloon.in	saloon.out
3 10 0 0 10 1 1 10 2 1	10 20 10 40 10 2
5 1 0 100 2 0 0 2 1 0 2 2 3 2 3 0	1 20 2 20 2 1 2 40 2 3

Задача D. Постфиксная запись

Имя входного файла:	`postfix.in`
Имя выходного файла:	`postfix.out`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $A\ B\ +$. Запись $B\ C\ +\ D\ *$ обозначает привычое нам $(B + C) * D$, а запись $A\ B\ C\ +\ D\ *\ +$ означает $A + (B + C) * D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение.

Формат входного файла

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции $+$, $-$, $*$. Строка содержит не более 100 чисел и операций.

Формат выходного файла

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю меньше 2^{31}.

Примеры

postfix.in	postfix.out
8 9 + 1 7 - *	-102

Задача E. Баржа

Имя входного файла:	`ship.in`
Имя выходного файла:	`ship.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На барже располагается K грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10 000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь не трогая остальных. Будем называть такие бочки крайними.

Изначально баржа пуста. Затем она последовательно проплывает через N доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более P бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.

Формат входного файла

В первой строке три целых числа N, K и P ($1 \le N, K, P \le 100\,000$). Далее следует N строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «$+\ A\ B$», где A — номер отсека, в который помещается бочка, а B — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «$-\ A\ B$», где A — номер отсека, из которого извлекается бочка, а B — номер ожидаемого вида топлива.

Формат выходного файла

Вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржой маршрута, либо вывести слово «Error» в противном случае.

Примеры

ship.in	ship.out
10 1 5 + 1 1 + 1 2 + 1 3 + 1 4 + 1 5 - 1 5 - 1 4 - 1 3 - 1 2 - 1 1	5