

Задача А. Скобки

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

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

Формат входного файла

В единственной строке входного файла записано подряд N скобок ($1 \leq N \leq 255$).

Формат выходного файла

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

Пример

| <code>brackets.in</code> | <code>brackets.out</code> |
|--------------------------|---------------------------|
| <code>([])</code> | YES |
| <code>(({}))</code> | NO |

Задача В. Постфиксная запись

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

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 привычное 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} .

Пример

| <code>postfix.in</code> | <code>postfix.out</code> |
|----------------------------|--------------------------|
| <code>8 9 + 1 7 - *</code> | -102 |

Задача С. Парикмахерская

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

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 характеристика, называемая *степенью нетерпения*. Она показывает сколько человек может максимально находиться 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 ухода 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 ухода 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.

Пример

| <code>saloon.in</code> | <code>saloon.out</code> |
|------------------------|-------------------------|
| 3 | 10 20 |
| 10 0 0 | 10 40 |
| 10 1 1 | 10 2 |
| 10 2 1 | |