

## Задача А. Результаты олимпиады

Имя входного файла: olymp.in  
Имя выходного файла: olymp.out  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

$N$  участников олимпиады получили уникальные номера от 1 до  $N$ . В результате решения задач на олимпиаде каждый участник получил некоторое количество баллов (целое число от 0 до 600). Известно, кто сколько баллов набрал.

Требуется перечислить участников олимпиады в порядке невозрастания набранных ими баллов.

### Формат входного файла

В первой строке входного файла содержится число  $N$  ( $0 \leq N \leq 101$ ). Далее записаны через пробел  $N$  чисел — количество набранных участниками баллов (1-е число — это баллы, набранные участником номер 1, 2-е — участником номер 2 и т.д.).

### Формат выходного файла

В выходной файл следует вывести  $N$  чисел — номера участников в порядке невозрастания набранных ими баллов (участники, набравшие одинаковое количество баллов могут быть выведены в любом порядке).

### Примеры

olymp.in	olymp.out
5 100 312 0 312 500	5 2 4 1 3

## Задача В. 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 ( $1 \leq N \leq 255$ ).

### 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 С. Воздушные шары

Имя входного файла: balloons.in  
Имя выходного файла: balloons.out  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

Несколько недель назад у Васи был день рождения. На этот праздник он получил большое количество воздушных шаров. Вася заметил, что каждый из шаров сдувается. Он заинтересовался в каком порядке шарики сдуются. Для этого он привлёк своего папу с его оборудованием. Про каждый шарик известен его исходный объем  $c_i$  (единиц объема) и скорость, с которой шарик сдувается,  $v_i$  единиц объема в единицу времени.

### Формат входного файла

В первой строке входного файла записано число  $N$  ( $1 \leq N \leq 1000$ ) — число шариков, полученных Васей на день рождения. Следующие  $N$  строк содержат по два числа,  $i + 1$  строка входного файла содержит изначальный объем  $i$ -ого шарика  $c_i$  и скорость сдувания шарика  $v_i$ . ( $1 \leq c_i, v_i \leq 100000$ )

### Формат выходного файла

Выходной файл в единственной строке должен содержать порядок, в котором шарики сдуются. Если какие-то шарики сдуются в одно и тоже время, то выведите их в любом порядке.

### Примеры

balloons.in	balloons.out
2 3 1 2 1	2 1
4 2 2 3 1 5 3 4 2	1 3 4 2

## 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 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  $2^{31}$ .

### Примеры

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

## Задача Е. 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 характеристика, называемая *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 выведете  $N$  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 20
10 0 0	10 40
10 1 1	10 2
10 2 1	
5	1 20
1 0 100	2 20
2 0 0	2 1
2 1 0	2 40
2 2 3	2 3
2 3 0	