

Задача А. Правильная круглоскобочная последовательность

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

Рассмотрим последовательность, состоящую из круглых скобок. Программа должна определить, является ли данная скобочная последовательность правильной.

Пустая последовательность является правильной. Если A — правильная, то последовательность (A) — правильная. Если A и B — правильные последовательности, то последовательность AB — правильная.

Формат входных данных

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

Формат выходных данных

Если данная последовательность правильная, то программа должна вывести строку *yes*, иначе строку *no*.

Примеры

стандартный ввод	стандартный вывод
()()	yes
(	no

Задача В. Правильная скобочная последовательность

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

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа должна определить, является ли данная скобочная последовательность правильной.

Пустая последовательность является правильной. Если A — правильная, то последовательности (A) , $[A]$, $\{A\}$ — правильные. Если A и B — правильные последовательности, то последовательность AB — правильная.

Формат входных данных

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

Формат выходных данных

Если данная последовательность правильная, то программа должна вывести строку *yes*, иначе строку *no*.

Примеры

стандартный ввод	стандартный вывод
<code>() []</code>	<code>yes</code>
<code>([])</code>	<code>no</code>

Задача С. Постфиксная запись

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

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. От вас требуется вычислить выражение, записанное в таком виде.

Формат входных данных

В первой строке входного файла записано число n — суммарное количество операций и чисел в выражении. В следующих n строках записан очередной элемент выражения в следующем формате:

- 0 x — очередной элемент является числом x ;
- 1 — очередной элемент является знаком сложения;
- 2 — очередной элемент является знаком вычитания;
- 3 — очередной элемент является знаком умножения;

Гарантируется, что никакие промежуточные вычисления не превзойдут $2 \cdot 10^9$.

Формат выходных данных

Необходимо вывести значение записанного выражения.

Пример

стандартный ввод	стандартный вывод
7 0 8 0 9 1 0 1 0 7 2 3	-102

Замечание

Пример соответствует выражению $8\ 9 +\ 1\ 7 - *$, которое равно $(8 + 9) * (1 - 7) = -102$

Задача D. Стек с минимумом

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

Вам дано q запросов трех типов:

- «1 x » — добавить число $0 \leq x \leq 10^{18}$ на вершину стека.
- «2» — вывести на экран и удалить верхний элемент стека. Гарантируется, что стек в данный момент времени не пуст.
- «3» — вывести на экран минимальное число стека. Гарантируется, что стек в данный момент времени не пуст.

Формат входных данных

В первой строке вводится одно число $1 \leq q \leq 10^6$ — количество запросов.

В последующие q строках вводятся запросы в выше описанном формате.

Формат выходных данных

Для каждого запроса второго и третьего типа выведите по одному числу.

Пример

стандартный ввод	стандартный вывод
5	2
1 3	2
1 2	3
3	
2	
3	

Задача Е. Очередь с минимумом

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

Вам дано q запросов трех типов:

- «1 x » — добавить число $0 \leq x \leq 10^{18}$ в конец очереди.
- «2» — вывести на экран и удалить первое число из очереди. Гарантируется, что очередь в данный момент времени не пуста.
- «3 i » — вывести на экран i -е число очереди. Гарантируется, что $1 \leq i \leq len$, где len — это размер очереди в данный момент времени.
- «4» — вывести на экран минимальное число очереди. Гарантируется, что очередь в данный момент времени не пуста.

Формат входных данных

В первой строке вводится одно число $1 \leq q \leq 10^6$ — количество запросов.

В последующие q строках вводятся запросы в выше описанном формате.

Формат выходных данных

Для каждого запроса второго, третьего и четвёртого типа выведите по одному числу.

Пример

стандартный ввод	стандартный вывод
6	2
1 2	2
1 3	3
4	3
2	
3 1	
4	

Задача F. Шарики

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

В одной компьютерной игре игрок выставляет в линию шарiki разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет «уничтожено». Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Формат входных данных

Сначала вводится количество шариков в цепочке $1 \leq n \leq 10^5$, а затем цвета шариков $0 \leq c_i \leq 9$.

Формат выходных данных

Требуется вывести количество шариков, которое будет «уничтожено».

Пример

стандартный ввод	стандартный вывод
5 1 3 3 3 2	3

Задача G. Сплоченность в IT

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

Совет топ-менеджеров одной известной IT компании решил, что для повышения эффективности работы необходимо улучшить взаимодействие между командами. Для этого в пятницу было организовано мероприятие по сплочению сотрудников. На мероприятие собрали сотрудников из 26 команд, каждая из которых обозначалась уникальной латинской буквой от 'a' до 'z'. Затем всех сотрудников, пришедших на мероприятие, выстроили в ряд. Для каждого сотрудника записали букву, соответствующую его команде, в результате чего получилась строка длины n , состоящая из латинских строчных символов.

Сотрудники из одной команды уже знакомы друг с другом, поэтому целью мероприятия стало познакомить людей из разных команд. Поскольку сотрудников было слишком много, решено было знакомить только тех программистов, которые являются соседями в ряду и работают в соседних командах. Соседними командами являются те, чьи буквы расположены подряд в алфавите. Например, 'a' и 'b', 'e' и 'd' — это пары соседних команд, а 'z' и 'a', 'd' и 'f', 't' и 't' соседними не считаются.

Для того чтобы процесс знакомства был более организованным и последовательным, было решено следующее: среди всех соседей, которые соответствуют условию для знакомства, выбиралась первая подходящая пара. Затем один из сотрудников этой пары исключался из ряда, а остальные сдвигались, занимая его место. При этом исключался сотрудник с буквой, которая идет позже в алфавите — то есть из пары удалялся сотрудник с максимальной буквой, а оставался тот, чья буква идет раньше в алфавите. Это правило позволяло сохранять упорядоченность строки и помогало равномерно знакомить сотрудников из разных команд.

Все детали были продуманы, теперь осталось понять, какая последовательность сотрудников останется в конце, для этого к вам обратились за помощью.

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Далее следуют описания наборов.

В первой строке дано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество сотрудников на мероприятии.

Во второй строке вводится строка из n латинских строчных символов $s_1 s_2 \dots s_n$ — последовательность команд сотрудников в ряду.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите команды оставшихся людей в ряду.

Пример

стандартный ввод	стандартный вывод
5	be
5	f
bdcef	daz
3	a
hgf	aea
7	
dbbdcaz	
4	
cbab	
8	
abebbcab	

Замечание

В первом наборе входных данных ряд меняется следующим образом : ‘bdcef’ → ‘bcef’ → ‘bef’ → ‘be’

В третьем наборе входных данных ряд меняется следующим образом : ‘dbbdcaz’ → ‘dbbcaz’ → ‘dbbaz’ → ‘dbaz’ → ‘daz’

Задача Н. Марсианская парикмахерская

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Пока я не поспал, «сегодня» не наступило

мистер Грин

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).

Также 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, также считается находящимся в очереди.

Требуется для каждого клиента указать время его выхода из парикмахерской.

Формат входных данных

В первой строке вводится натуральное число N , не превышающее 10^5 — количество клиентов.

В следующих N строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 16 000 000, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 10^5) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

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 выходных данных

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 нужно считать, что время его ухода равно времени прихода.

Примеры

стандартный ввод	стандартный вывод
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	

Задача I. Сумма Элементов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

У вас есть изначально пустой массив. Вам нужно обрабатывать запросы двух видов:

- Вид 1 x c : вам нужно добавить в конец массива c элементов, каждый из которых равен x
- Вид 2 c : вам нужно найти сумму первых c элементов массива, а затем удалить их. Гарантируется, что во время такого запроса в массиве находится хотя бы c элементов.

Формат входных данных

В первой строке вводится одно число Q ($1 \leq Q \leq 2 \cdot 10^5$) — число запросов.

Затем в следующих Q строках. Для запросов первого вида выполняется $0 \leq x \leq 10^9, 1 \leq c \leq 10^9$, а для запросов второго вида — $1 \leq c \leq 10^9$.

Все числа во входных данных — целые.

Формат выходных данных

Выведите ответы на все запросы второго типа в том порядке, в котором они идут во входных данных.

Примеры

стандартный ввод	стандартный вывод
4 1 2 2 2 1 1 4 5 2 3	2 10
2 1 1000000000 1000000000 2 1000000000	1000000000000000000

Замечание

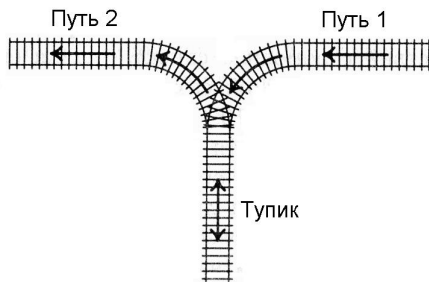
Покажем, как будет меняться массив в первом примере:

- Первая операция — добавить два числа 2 в массив. После этой операции массив будет равен $[2, 2]$.
- Вторая операция — найти сумму первого числа в массиве и удалить его. Эта сумма равна 2, а массив после удаления будет равен $[2]$.
- Третья операция — добавить пять чисел 4 в массив. После этой операции массив будет равен $[2, 4, 4, 4, 4, 4]$.
- Четвертая операция — найти сумму первых трех чисел в массиве и удалить их. Эта сумма равна $2 + 4 + 4 = 10$, а массив после удаления будет равен $[4, 4, 4]$.

Задача J. Сортировка вагонов

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

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.



Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика).

Формат входных данных

Вводится число N — количество вагонов в поезде ($1 \leq N \leq 2000$). Далее идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N , каждое из которых встречается ровно один раз.

Формат выходных данных

Если сделать так, чтобы вагоны шли в порядке от 1 до N , считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите действия, которые нужно проделать с поездом. В первой строке выведите количество действий, а затем сами действия. Каждое из них описывается двумя числами: типом и количеством вагонов:

- если нужно завезти с пути 1 в тупик K вагонов, должно быть выведено сначала число 1, а затем — число K ($K \geq 1$),
- если нужно вывезти из тупика на путь 2 K вагонов, должно быть выведено сначала число 2, а затем — число K ($K \geq 1$).

Если возможно несколько последовательностей действий, приводящих к нужному результату, выведите любую из них.

Если выстроить вагоны по порядку невозможно, выведите одно число 0.

Примеры

стандартный ввод	стандартный вывод
3 3 2 1	2 1 3 2 3
4 4 1 3 2	4 1 2 2 1 1 2 2 3

Задача К. Гоблины и очереди 2: Electric Boogaloo

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

Гоблины Мглистых гор очень любят ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Формат входных данных

В первой строке входных данных записано число N ($1 \leq N \leq 10^6$) — количество запросов. Следующие N строк содержат описание запросов в формате:

- $+ i$: гoblin с номером i ($1 \leq i \leq N$) встает в конец очереди.
- $* i$: привилегированный гoblin с номером i встает в середину очереди.
- $-$: первый гoblin из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Формат выходных данных

Для каждого запроса типа $-$ программа должна вывести номер гоблина, который должен уйти к шаманам.

Примеры

стандартный ввод	стандартный вывод
7 + 1 + 2 - + 3 + 4 - -	1 2 3
2 * 1 + 2	

Задача L. Постфиксная запись 2: Польша наносит ответный удар

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

Вы уже знаете, как устроена постфиксная запись выражений. Теперь представьте, что у вас есть бесконечноядерный компьютер, который за секунду может выполнить сколько угодно операций. Это значит, что выражение $(1+2)*(3+4)-6$ (которое в постфиксной записи выглядит как $1\ 2\ +\ 3\ 4\ +\ *\ 6\ -$) данный компьютер способен вычислить за три секунды: за первую секунду он параллельно вычислит обе суммы, за вторую, получив значение обоих сомножителей, вычислит произведение, а за третью произведёт вычитание. Более формально: любую операцию можно начинать выполнять сразу, как только оба её операнда уже вычислены.

Дано выражение в обратной польской записи. Определите, за какое наименьшее время может быть вычислено его значение.

Формат входных данных

В первой строке входного файла записано число n — суммарное количество операций и чисел в выражении ($1 \leq n \leq 10^5$). В следующих n строках записан очередной элемент выражения в следующем формате:

- 0 x — очередной элемент является числом $0 \leq x \leq 10^9$;
- 1 — очередной элемент является знаком сложения;
- 2 — очередной элемент является знаком вычитания;
- 3 — очередной элемент является знаком умножения.

Формат выходных данных

Необходимо вывести одно число — ответ на вопрос задачи.

Пример

стандартный ввод	стандартный вывод
9 0 1 0 2 1 0 3 0 4 1 3 0 6 2	3

Задача М. Перлы и конвертер

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Перлы — это мирная и первобытная раса, которая по вине человечества почти вымерла, а её оставшиеся представители дрейфовали по космосу. Прибыв на Альфу перлы познакомились с Вальерианом и Лорелин и смогли наконец-то обзавестись конвертером жемчужин.

Конвертер — миленький зверек, который производит жемчужины k различных цветов. Для запуска двигателя космического корабля перлам нужен набор из k различных по цвету жемчужин. Конвертер производит одну жемчужину в секунду. Для эффективной работы двигателя нужно, чтобы в каждом наборе для любой пары жемчужин выполнялось условие, что разница во времени между появлением этих жемчужин не превосходит m секунд. Каждая жемчужина может входить только в один набор.

Конвертер произвел n жемчужин и устал. Помогите перлам узнать, наибольшее возможное число наборов жемчужин, которые они смогут собрать из имеющихся жемчужин.

Формат входных данных

В первой строке содержатся три целых числа n , m , k — количество жемчужин, произведенных конвертером, максимальный промежуток времени между появлением каждой пары жемчужин в одном наборе и количество различных цветов жемчужин соответственно ($1 \leq m \leq n \leq 10^5$, $1 \leq k \leq 10^5$).

В следующей строке содержатся n целых чисел a_i — цвет i -й появившейся жемчужины ($1 \leq a_i \leq k$).

Формат выходных данных

В первой строке выведите одно число x — наибольшее возможное число наборов жемчужин, которые перлы смогут собрать из имеющихся жемчужин.

В следующих x строках выведите по k целых чисел d_{ij} — номера жемчужин, входящих в i -й набор ($1 \leq d_{ij} \leq n$).

Если подходящих ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
6 2 3 1 2 2 1 3 3	1 4 3 5
2 1 2 1 1	0
5 2 3 1 2 2 2 3	0