

Задача 1. Речные гонки

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

Егор и Петр участвуют в речной гонке на лодках: участники одновременно стартуют в пункте **A** и должны проплыть против течения реки в пункт **B**. Тот, кто приплывет первым, объявляется победителем. В результате жеребьевки Егору и Петру выпало участвовать в одном заплыве. Нам известны скорости движения лодок ребят и скорость течения реки. Ваша задача – определить, кто же из них победит.

Формат входного файла

В первой строке ввода содержатся три целых числа: **V1** – скорость лодки Егора, **V2** – скорость лодки Петра и **V3** – скорость течения реки ($0 \leq V1, V2, V3 \leq 10^6$).

Формат выходного файла

Выведите одну строку, содержащую следующий текст:

- "EGOR", если первым приплывет Егор;
- "PETR", если первым приплывет Петр;
- "RIVER", если победителя определить не удалось.

Пример входных и выходных файлов

<code>river.in</code>	<code>river.out</code>
10 5 1	EGOR
4 5 2	PETR
6 6 2	RIVER

Задача 2. Сеть

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

Для проведения олимпиады организаторы планируют объединить компьютеры участников в сеть. Из сетевого оборудования в наличии есть N коммутаторов и неограниченное количество сетевых кабелей. Коммутатор с номером i ($1 \leq i \leq n$) характеризуется числом a_i — количеством портов в этом коммутаторе.

Организаторы могут соединить кабелем либо два коммутатора, либо два компьютера, либо коммутатор и компьютер. Каждый коммутатор может быть соединен кабелями не более чем с a_i устройствами (коммутаторами или компьютерами), каждый компьютер — не более чем с одним.

Два компьютера могут обмениваться данными, если от одного из них до другого можно добраться по кабелям, возможно, пройдя при этом по цепочке из коммутаторов. Организаторы хотят построить сеть таким образом, чтобы каждые два компьютера могли обмениваться данными.

Какое максимальное количество компьютеров организаторы могут объединить в сеть, используя имеющиеся коммутаторы?

Формат входного файла

В первой строке входного файла находится одно число N — количество коммутаторов, имеющих у организаторов ($0 \leq N \leq 10^5$).

Во второй строке файла находится N чисел a_i — количество портов в коммутаторе с номером i ($1 \leq a_i \leq 10^9$, $1 \leq i \leq N$).

Формат выходного файла

Выведите единственное число — максимальное количество компьютеров, которое удастся объединить в сеть, используя имеющиеся коммутаторы.

Пример входных и выходных файлов

<code>network.in</code>	<code>network.out</code>
3 10 4 5	15
2 1 10	10
2 3 10	11

Задача 3. Большое число

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

Дано целое число N , состоящее из четного количества десятичных цифр. Над ним последовательно производятся следующие действия:

1. цифры числа разделяются на две равные половины;
2. левая и правая половины разворачиваются, то есть порядок следования цифр меняется на противоположный;
3. аналогичные действия выполняются для частей числа без первой и последней цифры, и так далее.

Когда останется последняя цифра первой половины числа и первая — второй, процесс останавливается, так как разворачивать их не имеет смысла.

Рассмотрим пример. Пусть $N = 1234567890$. Тогда в процессе выполнения указанных действий будет получена следующая цепочка: 5432109876, 5123478906, 5143298706, 5142389706. Ваша задача — узнать результат последовательности указанных преобразований.

Формат входного файла

Входной файл содержит единственное число N (не более 10000 цифр). Допускаются нули в начале записи числа.

Формат выходного файла

Выходной файл должен содержать единственное число — результат применения всех действий.

Пример входных и выходных файлов

<code>bignum.in</code>	<code>bignum.out</code>
1234567890	5142389706
000123	000231
012039	201390

Задача 4. Ох, уж эти скобки

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

Математическое выражение записано в виде произведения:

$$(\pm a_2 x^2 \pm a_1 x \pm a_0) \cdot (\pm b_2 x^2 \pm b_1 x \pm b_0) \cdot (\pm c_2 x^2 \pm c_1 x \pm c_0) \dots$$

Внутри каждой из N скобок произведения находится выражение вида:

$$\pm a_2 x^2 \pm a_1 x \pm a_0,$$

где хотя бы один из коэффициентов a_i не равен нулю (b_i , c_i и т. д., аналогично). Требуется составить программу, которая перемножает выражения в скобках и выводит полученную функцию в виде многочлена с приведенными по степеням x слагаемыми, то есть в виде:

$$\pm q_{2N} x^{2N} \pm q_{2N-1} x^{2N-1} \pm \dots \pm q_3 x^3 \pm q_2 x^2 \pm q_1 x \pm q_0.$$

Формат входного файла

В первой строке входного файла находится число N ($1 \leq N \leq 6$).

Во второй строке находится выражение из N пар скобок. Внутри каждой пары скобок находится выражение в виде « $\pm a_2 x^2 \pm a_1 x \pm a_0$ », где « \pm » — это или знак «+», или знак «-». Значение каждого из коэффициентов a_i , b_i , c_i и т.д. не превышает 10. Выражение составлено по следующим правилам:

1. Всё выражение записывается, начиная со старшей степени переменной по убыванию степеней.
2. Если какой-то коэффициент равен нулю, то этот коэффициент и соответствующий ему x опускаются в записи вместе с арифметическим знаком. Исключением является случай, когда все коэффициенты равны нулю. В этом случае вместо всего выражения указывается единственный коэффициент 0.
3. Если $a_i = \pm 1$ и $i > 0$, то единица перед соответствующим ему x не ставится.
4. Если первый отличный от нуля коэффициент положителен, то знак «+» перед ним опускается.
5. В выражении отсутствуют пробельные символы (пробел, табуляция) и знаки умножения.

Формат выходного файла

В первой строке выходного файла выведите результат раскрытия скобок в исходном выражении в следующем формате:

$$\pm q_{2N} x^{2N} \pm q_{2N-1} x^{2N-1} \pm \dots \pm q_1 x \pm q_0.$$

Формат выражения должен полностью соответствовать описанию для входного файла. Скобки вокруг степеней ставить не нужно, они приведены здесь только для читабельности.

Пример входных и выходных файлов

<code>brackets.in</code>	<code>brackets.out</code>
1 (3x ² +2x-1)	3x ² +2x-1
2 (4x ² +3x+5) (2x ² +4x+1)	8x ⁴ +22x ³ +26x ² +23x+5
3 (-x+7) (x) (x ² +x+1)	x ⁴ +6x ³ +6x ² +7x
6 (1) (2) (3) (4) (5) (6)	720