

Лабораторная состоит из не особо идейных задач на реализацию, для того чтобы набить руку в программировании на C++. В некоторых задачах могут потребоваться простые применения контейнеров C++, надеюсь вы с ними справитесь.

Длительность лабораторной одна неделя. Каждая неправильная посылка засчитывается за один минус, каждая несданная задача за один делей. Если вы не сдали более двух задач, лабораторная не засчитывается.

Задача А. Покер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Существо сидит напротив и хищно улыбается. Еще бы, ставка слишком высока. А ведь только недавно... Вы отправились на подвиги, но вдруг все пошло не по плану. Коня сдох, меч сперли, напарник спился,... Умирая от голода Вы встретили гоблина, он даже согласился отдать Вам его вчерашний борщ, но только если Вы выиграете у него в его любимую игру. И вот, теперь, Вы пытаетесь выиграть в гоблинский покер у гоблина, задача почти не выполнимая. Но вы решили попробовать...

Правила этой версии покера крайне просты, есть n карт, Вам нужно ответить на m вопросов типа 'Есть ли среди данного набора карт данная комбинация?'.

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 52$) - количество карт на столе. В следующей строке через пробел содержатся n карт.

Карта вводится в следующем формате: пар символов - первый символ пары целое число от 2 до 9 или символы 'T', 'J', 'Q', 'K', 'A' обозначающие номинал карты, второй символ пары буква 'S', 'C', 'D', 'H'.

В третьей строке вводится единственное целое число m ($0 \leq m \leq 10^5$) - количество запросов.

В следующих m строках вводится по одной фразе из следующего списка:

High card <Card>

Pair of <Cost>

Two pair <Cost> and <Cost>

Set of <Cost>

Flush on the <Suit>

Straight start with <Cost>

Full-Hause of <Cost> and <Cost>

Care of <Cost>

Straight-Flush start with <Card>

Где <Card> - карта в формате описанном выше; <Suit> - масть карты - одно из четырех слов : Spades, Clubs, Diamonds, Hearts <Cost> - номинал карты - число от 2 до 9, либо символ 'T' - 10, 'J' - валет, 'Q' - дама, 'K' - король, 'A' - туз.

Гарантируется что для Straight-Flush и Straight вводится старшая карта пятерки. Для Full-Hause сначала вводится номинал для трех картЮ затем для двух.

Формат выходных данных

Для каждого запроса выведите в отдельной строке 'Yes' или 'No', 'Yes' если комбинация из соответствующего запроса есть среди карт, и 'No' в противном случае.

Примеры

стандартный ввод	стандартный вывод
10	Yes
8H 6D QC 8H 8D 9C 6D QH 5H 5C	Yes
5	No
Pair of 5	Yes
Flush on the Spades	Yes
Set of 8	
Two pair Q and 5	
Pair of 8	

Замечание

пики	Spades	S
крести	Clubs	C
буби	Diamonds	D
черви	Hearts	H

High card - старшая карта

Pair - пара карт одного номинала

Two pair - две пары карт, внутри пары карты одного номинала

Set - три карты одного номинала

Flush - пять карт одной масти

Straight - пять карт подряд идущего номинала

Full-Hause - три карты одного номинала + две карты другоно наминала (pair + set)

Care - четыре карты одного номинала

Straight-Flush - пять карт одной масти и подряд идущего номинала

Задача В. Роман и Тошибки

Имя входного файла: `broken-keyboard.in`
Имя выходного файла: `broken-keyboard.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Казённые ноутбуки ездят в ЛКШ уже очень давно, а потому многое пережили.

Рома тоже ездит в ЛКШ очень давно, а потому пережил ещё больше.

Очень старым Романом на очень старом ноутбуке открыт очень старый файловый менеджер MOOSE. Чтобы выбрать нужный файл в этом файловом менеджере, достаточно просто набрать его имя, при этом после каждого нажатия клавиши с буквой MOOSE переходит к первому в списке файлу, начинающемуся с набранной последовательности букв. Чтобы открыть файл, нужно выбрать его в файловом менеджере, а затем нажать клавишу Enter.

Так же, как с Романа сыплется песок, с ноутбуков сыплются клавиши. А именно, после первого нажатия клавиша отваливается и становится нерабочей. В файловом менеджере MOOSE отображается некоторый список файлов в лексикографическом порядке. Для каждого файла укажите, сможет ли его открыть Роман, если он не собирается трогать клавиши, отличные от клавиш с латинскими буквами и клавиши Enter. Стоит заметить, что Роман очень ленив, поэтому его интересует чисто теоретическая возможность открытия файлов, ни на какие клавиши он нажимать не будет.

Напомним, что строка s длины n лексикографически меньше строки t длины m , если выполняется одно из следующих условий:

- $n < m$ и при этом $s_i = t_i$ для всех i от 1 до n ;
- существует индекс $k \leq \min(n, m)$ такой, что $s_i = t_i$ для всех i от 1 до $k - 1$ и $s_k < t_k$.

Например, строка `abc` лексикографически меньше строки `b`, а строка `b` лексикографически меньше строки `babс`.

Формат входных данных

В первой строке входных данных содержится одно число n ($1 \leq n \leq 1000$) — количество файлов в директории, открытой файловым менеджером MOOSE.

В следующих n строках содержатся имена отображаемых в MOOSE файлов в лексикографическом порядке. Имена всех файлов содержат от одного до 100 символов и состоят из строчных букв латинского алфавита.

Формат выходных данных

Для каждого файла выведите «YES» (без кавычек), если Роман может открыть этот файл, и «NO» (без кавычек), если не может.

Примеры

<code>broken-keyboard.in</code>	<code>broken-keyboard.out</code>
5	YES
ab	YES
abc	YES
b	YES
babс	NO
babd	

Задача С. Тасовка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Тасование колоды карт происходит следующим образом. Колода разбивается на несколько частей перегородками, которые нумеруются по номеру стоящей после неё карты (карты нумеруются с нуля). Затем полученные таким образом блоки карт переставляются в обратном порядке (при этом внутри каждого блока, порядок карт сохраняется). Требуется по заданным значениям карт и заданными номерами перегородок, вывести порядок карт после одного такого тасования.

Формат входных данных

Во входном файле находятся две строки. В первой строке содержатся значения карт по порядку через пробел. При этом гарантируется, что значение каждой карты по модулю не более 10^{15} , а количество карт не более 10^6 . Во второй строке содержатся номера перегородок в порядке возрастания через пробел. Перегородки могут ставиться только между картами. Ни в каком промежутке между картами не может находиться более одной перегородки.

Формат выходных данных

В выходной файл надо вывести одну строку — полученную в результате тасовки последовательность значений карт.

Примеры

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

Задача D. Кроссворды

Имя входного файла: `crosswords.in`
Имя выходного файла: `crosswords.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Саламандер очень любит своих существ и поэтому решил скрасить им времяпровождение в чемодане. Он решил каждому существу дать по кроссворду, однако их много, и кроссвордов соответственно нужно тоже достаточное число. Саламандер решил, что для простоты каждый кроссворд будет состоять ровно из 4 слов — 2 по вертикали и 2 по горизонтали, а также у них будет ровно 4 попарных пересечения, образующих прямоугольник, возможно нулевой площади.

Саламандер уже придумал 4 различных слова для нового кроссворда, теперь он хочет знать количество различных кроссвордов, которые из них можно составить. Он считает, что два кроссворда различны, если их нельзя наложить друг на друга так, чтобы они полностью совпали. Помогите ему посчитать нужное ему количество способов.

Формат входных данных

В i -й из 4 строк содержится w_i — i -е слово, придуманное Саламандром ($2 \leq |w_i| \leq 30$). Гарантируется, что каждое слово состоит только из строчных букв латинского алфавита и что все слова попарно различны.

Формат выходных данных

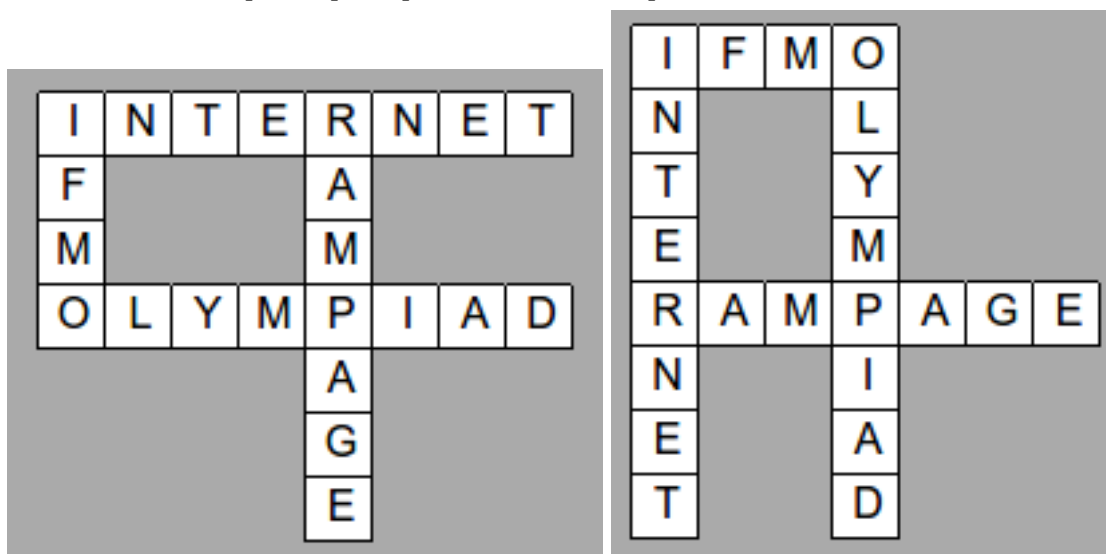
В единственной строке выведите количество способов составить кроссворд из данных слов. Учтите, что слова по горизонтали записываются слева направо, а слова по вертикали — сверху вниз.

Пример

crosswords.in	crosswords.out
internet ifmo rampage olympiad	2

Замечание

Оба возможных кроссворда представлены на картинках:



Задача Е. НОД

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

Формат входных данных

Входной файл содержит два натуральных числа a и b ($1 \leq a, b \leq 10^{18}$);

Формат выходных данных

В выходной файл выведите наибольший общий делитель этих двух этих чисел.

Примеры

<code>gcd.in</code>	<code>gcd.out</code>
3 5	1
3 9	3

Задача F. Многочлены

Имя входного файла: `polynomial.in`
Имя выходного файла: `polynomial.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Снежок нашел листок из старого учебника по математике. На нем перечислены t многочленов от двух переменных.

Теперь Снежок хочет их упростить. Помогите ему.

Во входном файле многочлены записаны в следующем формате:

- Многочлен не содержит пробелов и состоит только из арабских цифр и символов «n», «m», «+», «-», «*», «^».
- Многочлен состоит из нескольких слагаемых.
- Если слагаемое входит со знаком минус, то перед ним пишется «-».
- Если слагаемое входит со знаком плюс и это не первое слагаемое, то перед ним пишется «+».
- Каждое слагаемое можно представить в виде $a \cdot n^b \cdot m^c$, где a , b и c — целые неотрицательные числа (если не учитывать знак перед слагаемым), без ведущих нулей, не превышающие 10^5 .
- Слагаемое записывается в следующем формате:
 - Слагаемое является произведением чисел и степеней n и m .
 - Назовем числом последовательность подряд идущих цифр, слева и справа от которой стоят не цифры.
 - Перед каждым символом «^» находится переменная, то есть символ «n» или символ «m».
 - После каждого символа «^» находится число.
 - Таким образом, знак «^» обозначает, что переменная, стоящая перед ним, возводится в степень числа, стоящего после него. Назовем это множителем.
 - Если после символов «n» или «m» нет символа «^», то считается, что эта переменная в этом месте стоит в степени 1. Назовем каждую такую переменную множителем.
 - Все числа, которые не являются степенями (то есть, те, перед которыми не стоят символы «^»), назовем множителями.
 - По обе стороны от каждого символа «*» находятся множители.
 - Слагаемое это произведение всех его множителей.
 - Обращаем ваше внимание, что некоторые соседние множители могут не разделяться символом «*».

Для упрощенного многочлена должно выполняться следующее:

- Он записывается в таком же формате, как и исходный.
- Если многочлен тождественно равен нулю, нужно вывести «0».
- Если слагаемое представимо в виде $a \cdot n^b \cdot m^c$, то такое слагаемое должно быть записано как «an^bbm^c», где на место «a», «b» и «c» подставлены соответствующие числа.
- Для каждой пары a и b существует не более одного слагаемого, представимого в виде $k \cdot n^a \cdot m^b$.
- Если $n^a \cdot m^b$ входит в многочлен с коэффициентом 0, такое слагаемое писать не надо.
- Если $n^a \cdot m^b$ входит в многочлен с коэффициентом, по модулю равным 1, в таком слагаемом не нужно писать коэффициент.

- В слагаемом вида $a \cdot n^0 \cdot m^c$, переменную «n» писать не надо.
- В слагаемом вида $a \cdot n^1 \cdot m^c$, после переменной «n» не нужно писать степень.
- Для переменной «m» тоже выполняются предыдущие два пункта.
- Слагаемые в многочлене могут идти в любом порядке.

Формат входных данных

В первой строке задано целое число t — количество многочленов ($1 \leq t \leq 100$). В следующих t строках заданы многочлены. Гарантируется, что все числа в них не превосходят 10^5 .

Длины всех строк не превосходят 100.

Формат выходных данных

Для каждого многочлена выведите его в упрощенном виде.

Гарантируется, что в упрощенном многочлене все числа не превосходят 10^9 .

Пример

polynomial.in	polynomial.out
5	m-n+2n^2m
-n^2*m^1+n*nm3+mn-n+m-n*m	-2nm+5n^2m^2
n*n*m^0m^2*5-m*2n	23-100nm
-100n*m+23+0n^3	0
n-n	nm^2
100000*100000*100000*n^3*n^4*0+nm^2	

Замечание

Пояснение к примеру:

$$-n^2 \cdot m^1 + n \cdot nm3 + mn - n + m - n \cdot m = m - n + 2n^2m$$

$$n \cdot n \cdot m^0m^2 \cdot 5 - m \cdot 2n = -2nm + 5n^2m^2$$

$$-100n \cdot m + 23 + 0n^3 = 23 - 100nm$$

$$n - n = 0$$

$$100000 \cdot 100000 \cdot 100000 \cdot n^3 \cdot n^4 \cdot 0 + nm^2 = nm^2$$

Обращаем ваше внимание, что хотя в процессе вычисления значения коэффициента в первом слагаемом последнего многочлена оно и становится больше 10^5 , но в итоге это слагаемое равно $0 \cdot n^3 \cdot n^4$, значит для него ограничение на величину коэффициента тоже выполняется.

Задача G. Реверс

Имя входного файла: `reverse.in`
Имя выходного файла: `reverse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Переверните массив чисел. Без циклов. Без массивов. Без контейнеров.

Формат входных данных

В первой строке — число N . Далее в N строках указаны N чисел ($1 \leq N \leq 10000$), по одному числу на строке. Все числа по модулю не превышают 10^5 .

Формат выходных данных

Выведите N чисел — перевернутый массив.

Примеры

<code>reverse.in</code>	<code>reverse.out</code>
5 18086 53523 871 93415 50530	50530 93415 871 53523 18086

Задача Н. СУ-ДО-КУ

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

Бабушка Авдотья работает вахтером в школе. Поскольку у неё много свободного времени, она любит разгадывать sudoku. Как-то раз, проходя мимо, Леша увидел еще не решенное бабушкой sudoku размера 9×9 и попробовал решить его. Потратив на это весь день, он все же справился, правда до сих пор не уверен, правильно ли он это сделал. Поэтому он решил спросить бабушку Авдотью, правильно ли он заполнил sudoku. Бабушка очень занята просмотром передачи «Поле чудес» и не может сейчас проверить sudoku, поэтому просит вас помочь ей в этом.

Sudoku считается решенным правильно, если в каждой строке, каждом столбце и каждом специальном квадрате (квадраты одного цвета на картинке) встречаются все 9 цифр от 1 до 9.

5	8	1	2	4	9	7	6	3
7	4	3	8	6	1	2	9	5
2	6	9	7	3	5	4	8	1
8	9	6	4	5	2	3	1	7
3	2	7	9	1	6	5	4	8
4	1	5	3	7	8	9	2	6
1	5	4	6	9	3	8	7	2
9	3	2	1	8	7	6	5	4
6	7	8	5	2	4	1	3	9

Формат входных данных

На вход дается матрица 9×9 — решение sudoku.

Формат выходных данных

Вывести единственную строку — «Yes», если sudoku решено правильно, или «No» в противном случае (без кавычек).

Пример

стандартный ввод	стандартный вывод
5 8 1 2 4 9 7 6 3 7 4 3 8 6 1 2 9 5 2 6 9 7 3 5 4 8 1 8 9 6 4 5 2 3 1 7 3 2 7 9 1 6 5 4 8 4 1 5 3 7 8 9 2 6 1 5 4 6 9 3 8 7 2 9 3 2 1 8 7 6 5 4 6 7 8 5 2 4 1 3 9	Yes

Задача I. Странное число

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

Математик Петя снова складывает числа. На этот раз он решил, что число *странное*, если представимо в виде суммы чисел, в каждом из которых все цифры одинаковы. К примеру, число 100 — странное, потому что представимо как $99 + 1$, в каждом из которых все цифры одинаковы.

К сожалению, эта задача кажется ему сложнее, и он просит Вас о помощи. Подскажите Пете, является ли интересующее его число странным, и если да, то как разложить его на слагаемые из одинаковых цифр.

Формат входных данных

В единственной строке входных данных задано натуральное число $x < 10^{100}$. В числе x возможны ведущие нули.

Формат выходных данных

В первой строке выведите количество слагаемых из одинаковых цифр, сумма которых равна x . Во второй строке выведите сами эти числа через пробел.

Если возможных вариантов несколько, выведите любой из них. Если число x не странное, выведите в первой строке единственное число -1 .

Пример

стандартный ввод	стандартный вывод
13332	3 5555 5555 2222