



А. До конца пары

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Вася Пупкин уже третий год учится в X-OSPU (eXtreme Open Source Programming University). Ещё в прошлом году Вася перестал ходить на первые пары, а теперь он уж совсем редко показывается в университете. Настолько редко, что однажды, сидя на скучной лекции (на которую просто необходимо было прийти, если хочешь получить зачёт), он понял, что уже не помнит, когда заканчиваются пары.

Конечно, ходить на занятия чаще и постепенно снова запомнить — это не вариант. Вместо этого Вася решил доработать свою программу с расписанием занятий для RobotOS, которую он написал ещё на первом курсе. Просто показывать время конца пары и заставлять пользователя самостоятельно производить сложные вычисления в уме — это не хорошо, решил Вася, поэтому его программа должна сразу выводить, сколько времени осталось терпеть до конца пары. К сожалению, у Васи сейчас полный завал с долгами, а ещё и нужно обновлять свой смартфон до RobotOS 7.1 Nutella, так что он просит вас помочь с новым функционалом.

На вход вашего модуля подаётся текущее время и время окончания пары. Модуль должен вернуть количество времени, оставшегося до конца пары. Ночью занятия в X-OSPU не ведутся, поэтому переходов чере полночь точно не будет.

Входные данные

В первой строке указано текущее время, а во второй — время окончания пары. И то, и другое время дано в формате ЧЧ:ММ, то есть часы и минуты с ведущими нулями, разделённые двоеточием. Часы, как вы можете знать, могут находиться в диапазоне от 0 до 23 включительно, а минуты — от 0 до 59 тоже включительно. Гарантируется, что текущее время не больше, чем время окончания пары.

Выходные данные

Выведите количество времени, оставшегося до конца пары в формате ЧЧ:ММ, то есть количество часов и минут с ведущими нулями, разделённые двоеточием.

Пример

Входные данные	Результат
08:42 09:50	01:08
12:37 13:15	00:38

В. Странный девайс

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Шел 2025-ый год. Третьеклассник Коля сидит на уроке по математике в начальной школе. И уже третий год задается вопросом целесообразности пребывания в школе «он-сайт». Ведь можно же слушать лекции не вставая с дивана – смотря трансляцию по смартфону. А так его каждый день заставляют общаться с людьми, двигаться – не по современному это!

И вот в очередных попытках найти ответ на свой вопрос, парень, по привычке, листает новостную ленту в поисках смешных картинок с котиками. Внезапно ему попадает на глаза картинка со странным устройством, именуемым «Nokia 3310». Немного поискав в интернете, он понял, что подобными телефонами когда-то пользовались люди. «Но как? — спрашивает Коля. Ни тебе Wi-Fi, ни 7g, даже Full HD камеры нет. Да какой из него телефон – жалкое подобие. Как с него вообще могли звонить?». Так же Коля обнаружил странные кнопки с буквами на фронтальной части телефона. «С их помощью, оказывается, когда-то набирали сообщения?!», — удивленно думает про себя парень.



Поскольку на уроке было очень скучно, Коля решил себя занять и посчитать, сколько времени те первобытные люди тратили на набор одного сообщения. Он предположил, что телефон имеет K кнопок, а сообщение состоит только из маленьких латинских литер. Для примера Коля взял случайную строку из новостной ленты. Парень хочет так распределить буквы по кнопкам, чтобы алфавит сохранил свою последовательность, а количество нажатий на кнопки для набора сообщения было минимальным. Другими словами, Коле необходимо оптимальным образом сопоставить каждой кнопке определенный набор букв (возможно пустой), при этом чтобы, если выписать все буквы в порядке следования кнопок, мы получили исходный алфавит.

Коля не очень способный парень (так как смотрит котиков на уроке математики), в связи с чем просит вашей помощи.

P.S. Если вдруг вы одноклассник Коли и не знаете, как набирать текст на кнопочном телефоне, загляните в примечание.

Входные данные

В первой строке находится строка S , длина которой не превышает 10^5 , состоящая из маленьких латинских букв, — сообщение, над которым проводятся вычисления. В следующей строке содержится число K ($1 \leq K \leq 10$) — количество кнопок на телефоне.

Выходные данные

Выведите в единственной строке одно число – минимальное количество нажатий на кнопки телефона для набора сообщения.



Пример

Входные данные	Результат
lol 1	39
azaza 2	5
smartphone 9	11

В первом тесте у нас единственный выбор – весь алфавит в одной кнопке. Во втором тесте распределение может быть следующим: (abcdefghijklmnopqrstuvwxy)(z). В третьем тесте распределение может быть таким: (abcd)(efg)(hijkl)(m)(n)(o)(pq)(r)(stuvwxyz).

Примечание

При первом нажатии кнопки на экране появляется первая буква, обозначенная на кнопке. При повторном нажатии она сменяется на вторую букву. Если после появления последней буквы кнопки мы еще раз на неё нажмем – снова появится первая буква.

Стоит обратить внимание на то, что в данной задаче мы пренебрегаем переключением между буквами. Это означает, что после набора нужной буквы все нажатия сбрасываются. Посмотрите пример ниже для лучшего понимания.

Предположим, что телефон имеет 3 кнопки, а алфавит состоит из букв abcdefg. Допустим, что буквы распределены по кнопкам следующим образом: (abc)(def)(g). Для набора текста bacefg мы потратим 2+1+3+2+3+1 нажатий, а последовательность нажатий будет выглядеть следующим образом: (abc)(abc)|(abc)|(abc)(abc)(abc)|(def)(def)|(def)(def)(def)|(g). Обратите внимание на то, что третьим нажатием мы получили строку ba, а не c, так как нужную букву b мы уже получили на предыдущем шаге, и все нажатия сбросились.



С. Федор и аконкагуажцы

Ограничение по времени: для C/C++ — 1 секунда, для Java — 4 секунды
Ограничение по памяти: 256МБ

Друзья Федора с планеты Аконкагуа прилетели поздравить его с успешной сдачей сессии. Согласно традициям, жители этой планеты становятся вокруг виновника торжества и берутся за руки (никто не знает, как правильно называть их конечности). Два аконкагуажца могут взяться за руки только в том случае, если их рост отличается ровно на 1 аконкагуажский метр (это настолько странная единица измерения, что существуют аконкагуажцы с отрицательным ростом).

N аконкагуажцев взялись за руки и окружили Федора. Федор должен выбрать двух аконкагуажцев и загадать желание (считается, что именно эти два аконкагуажца помогут исполнить желание). Федор убежден, что желание сбудется только в том случае, если он выберет двух аконкагуажцев с одинаковым ростом. Кроме того, Федор считает, что, чем дальше друг от друга стоят аконкагуажцы, тем больше вероятность исполнения желания. В данном случае, расстояние между двумя аконкагуажцами определяется как минимальное из двух (в порядке обхода по и против часовой стрелки) количеств аконкагуажцев, стоящих между ними.

Помогите Федору выбрать двух аконкагуажцев одинакового роста, максимально удаленных друг от друга.

Входные данные

В первой строке дано число N ($4 \leq N \leq 3 \cdot 10^5$) — количество аконкагуажцев, окруживших Федора. Во второй строке дано N чисел A_1, A_2, \dots, A_N ($-10^9 \leq A_i \leq 10^9$) — росты аконкагуажцев. Росты перечислены в порядке обхода по часовой стрелке. Гарантируется, что $|A_i - A_{i-1}| = 1$ для всех i от 2 до N . Кроме того, $|A_1 - A_N| = 1$.

Выходные данные

Выведите два числа — номера аконкагуажцев согласно Фединым предрассудкам. Если ответов несколько, выведите любой.

Пример

Входные данные	Результат
4 1 2 3 2	2 4
6 4 5 6 5 6 5	2 4
10 525 526 525 524 525 524 525 526 525 524	1 5



D. Повреждённый сервер

Ограничение по времени: для C/C++ — 4 секунды для Java — 10 секунд

Ограничение по памяти: 256МБ

Представьте, что вы администратор сервера. И в первый же ваш рабочий день работоспособность рабочих станций, за которые он отвечает, пытаются нарушить.

Сервер обслуживает N ($1 \leq N \leq 10^4$) рабочих станций, пронумерованных от 1 до N . Злоумышленники хотят привести в негодность каждую из них. Система безопасности на станциях не безупречна, но вы знаете, что каждая рабочая станция может выдержать атаку на протяжении не менее 15 секунд, прежде чем её можно будет взломать. Это означает, что у вас есть минимум 15 секунд между взломами, чтобы успеть сделать что-то полезное.

Всё, чему вы успели научиться — это отправка запросов на сервер. У вас в распоряжении два вида запросов (на каждый из которых уходит по 1 секунде времени):

- «? L R» — спросить у сервера, сколько рабочих станций еще работают в промежутке от L до R . В ответ сервер пришлёт число K — искомое количество.
- «! V» — записать на стороне сервера число V .

Вы понимаете, что работоспособность рабочих станций не спасти. Но сделать что-то полезное, чтобы вас не отругало начальство за безделье, стоит. Вы решили записать на сервере порядок взлома станций. Иными словами, вы хотите получить перестановку A из N чисел, где A_i — номер рабочей станции, которую взломали i -ой по счету. Для этого у вас есть два вида запросов и совсем мало времени...

Протокол взаимодействия

Данная задача является интерактивной. Это означает, что вместо считывания статических входных данных вы взаимодействуете с проверяющей программой.

В самом начале ваша программа должна считать число N — количество рабочих станций, за которые отвечает сервер. Затем вы можете любое количество раз отправлять на сервер запросы по следующей инструкции:

- Выведите текст запроса (с переводом строки в конце) в стандартный выходной поток
- Сбросьте буфер стандартного выходного потока
- Считайте ответ из стандартного входного потока (если запрос подразумевает ответ)

Чтобы сбросить буфер выходного потока, обычно можно выполнить следующее:

- В языке C и C++: `fflush(stdout)`, `cout.flush()` или `cout << endl`.
- В языке Java: `System.out.flush()`.

По окончании работы просто завершите работу своей программы. Нарушение протокола взаимодействия может привести к вердиктам `Presentation Error` (явное нарушение), `Wall Time Limit Exceeded` (скорее всего ваша программа бесконечно ждёт ответа на неправильно составленный запрос) или `Time Limit Exceeded` (если ваша программа ушла в бесконечный цикл после неудачного чтения ответа на запрос).

Пример

Для краткости минимальный промежуток между взломами заменен на 3 секунды.

Входной поток	Выходной поток	Комментарий	Прошло секунд
2		Вводим число N	0
	? 1 2	Запрашиваем сумму от 1 до 2	1



Входной поток	Выходной поток	Комментарий	Прошло секунд
2		Получаем ответ 2	1
	? 1 2	Запрашиваем сумму от 1 до 2	2
2		Получаем ответ 2	2
	? 1 2	Запрашиваем сумму от 1 до 2	3
1		Получаем ответ 1	3
	? 1 1	Запрашиваем сумму от 1 до 1	4
1		Получаем ответ 1	4
	! 2	Записываем число 2	5
	? 1 1	Запрашиваем сумму от 1 до 1	6
0		Получаем ответ 0	6
	! 1	Записываем число 1	7

Итоговая последовательность: 2, 1. Именно в таком порядке были взломаны рабочие станции.



Е. Зимние подарки

Ограничение по времени: 2 секунды

Ограничение по памяти: 256МБ

Собираясь отнести новогодние подарки для детей, Дед-Мороз подготовил N подарочных наборов, упакованных в пакеты. Каждый набор имеет свой вес W_i и свою ценность C_i с точки зрения детей.

Когда Дед-Мороз собирался укладывать подарки в свой мешок, оказалось, что есть еще одно большое яблоко весом WA . Было решено разделить яблоко на K равных частей и распределить эти части между подарочными пакетами произвольным образом. При этом вес некоторых пакетов увеличивается, а их ценность с точки зрения детей не изменилась.

После окончательной упаковки всех подарков Дед-Мороз должен решить (с Вашей помощью) такую задачу: выбрать из этого множества некоторое подмножество пакетов таким образом, чтобы их суммарная ценность была максимальной возможной. При этом надо учесть ограничение на суммарный вес предметов в мешке WS .

Входные данные

Первая строка содержит 2 целых числа, разделенные пробелом: N ($1 \leq N \leq 32$) — количество подарочных пакетов, и WS ($0 \leq WS \leq 8000$) — максимальный суммарный вес пакетов в мешке.

В каждой из следующих N строк содержится по 2 целых числа, разделенных пробелом: W_i ($0 \leq W_i \leq 1024$) — вес i -го пакета, и C_i ($0 \leq C_i \leq 1024$) — его стоимость.

Далее следует строка с двумя целыми числами, разделенными пробелом: WA ($0 \leq WA \leq 4000$) — вес яблока, и K ($1 \leq K \leq 32$) — число частей, на которые его разделили.

В каждой из следующих N строк содержится по одному целому числу P_i ($0 \leq P_i \leq K$, $P_1 + P_2 + \dots + P_N = K$) — количество частей яблока в i -ом пакете.

Выходные данные

Выведите одно целое число — максимальную возможную суммарную ценность уместившихся в мешок Деда-Мороза подарков.

Пример

Входные данные	Результат
5 15 6 5 7 3 3 1 2 3 5 6 6 3 1 0 1 0 1	11



Ф. А или В

Ограничение по времени: 15 секунд

Ограничение по памяти: для C/C++ — 15МБ, для Java — 25МБ

Думаете, считать строку и тут же вывести её — это очень просто? Что ж, на самом деле так и есть. А попробуйте тогда считать две строки и вывести одну из них. Ту из них, которая не равна третьей строке. Думаете, всё равно легко? Ну попробуйте!

Вам даны три строки: А, В и С. Известно, что $A \neq B$ и либо $C = A$, либо $C = B$. Если $C = A$, выведите строку В, иначе — выведите строку А.

Входные данные

В трёх строках входного потока даны строки А, В и С. Строки разделены символами конца строки, и после третьей строки тоже есть символ конца строки. Длины всех трёх строк находятся в диапазоне от 0 до 10^7 символов. Строки могут содержать маленькие и большие английские буквы, цифры, символы `. , ? : ; ! @ # $ % ^ & * () - _ + = [] { } \ | ' " < > / ` ~` и пробелы.

Учтите, что входной поток данных можно прочесть только один раз.

Выходные данные

Выведите требуемую строку (А или В) в выходной поток в том же виде, в котором она была подана на вход.

Учтите, что выходной поток нельзя ни читать, ни перезаписывать.

Пример

Входные данные	Результат
foo bar foo	bar
foobar baz baz	foobar

Примечание

Да-да, в этой задаче есть подвох. Всё ещё не видите? Ну что же, всего одна посылка — и вы сразу всё поймёте :-). Хотя... может и нет...



G. Реформа

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Очередная порция абсурда произошла на этой неделе в Берляндии. Людей хотят наглým образом обмануть, проведя очередную денежную реформу, смысл которой только в воровстве.

Реформа заключается в изменении денежной валюты. Использувавшиеся ранее бурли будут заменены на монеты различных номиналов. Заключается реформа в следующем:

- Чтобы получить монету номиналом 2^i , нужно предоставить a_i бурлей к обмену.
- Нельзя получить две монеты номиналов А и Б, где $A = 2 \cdot B$

Никита, один из немногих в своём окружении, понимает всю абсурдность грядущих изменений. Он вызвался помочь людям. А именно: подсказать окружающим как правильно нужно распределять бурли по номиналам, чтобы получить как можно больше монет за меньшее количество бурлей. Но порой стоимости номиналов распределены настолько не очевидным образом, что оптимальный расчет может занять целый вечер, за время которого курс успеет несколько раз измениться.

Никита слышал, что вы занимаетесь программированием. Помогите ему написать программу, которая по известной сумме номиналов монет определит минимальное количество бурлей, необходимых для обмена.

Входные данные

В первой строке дано число n ($1 \leq n \leq 61$) – количество номиналов, для которых известен курс. В следующих n строках содержатся по два числа, b_i и p_i ($0 \leq b_i \leq 60$ и $0 \leq p_i \leq 100$) – номинал и его курс соответственно. Это означает, что за монету номинала 2^{b_i} необходимо отдать p_i бурлей. Считайте, что если какого-то номинала нет во входных данных, то он раздается бесплатно. В последней строке содержится число x ($1 \leq x \leq 10^{18}$) – сумма номиналов монет, которую необходимо получить.

Выходные данные

В единственной строке выведите одно число – минимальное количество бурлей для обмена.

Пример

Входные данные	Результат
2 0 5 1 4 3	15
4 0 1 1 5 2 1 3 10 15	6

В первом примере необходимо купить три монеты номиналом 1. Во втором — мы покупаем три монеты номиналом 1 и три монеты номиналом 4.



Н. Федор и маршрутки

Ограничение по времени: 3 секунды

Ограничение по памяти: 256МБ

Отпраздновав сдачу сессии, Федор очень плохо выспался. Он совершенно забыл, что обещал приехать к своему другу — Никите. Федор живет возле остановки с номером 1, а Никита — возле остановки с номером N.

Всего в городе N остановок. По городу ездят M маршруток. Все они стартуют в одно время из остановки под номером один. После того, как маршрутка приезжает в конечную для нее остановку, она сразу же едет по этому же маршруту в обратную сторону, заканчивая его в остановке номер 1 и тут же начиная новый круг. Так маршрутки ездят достаточно долго — до глубокой ночи.

Проснувшись, Федор осознал, что у него нет времени медлить. Он хочет приехать к своему другу Никите как можно быстрее. Помогите ему посчитать минимальное время, за которое он может добраться. Можно пересаживаться с одной маршрутки на другую сколько угодно раз. Время ожидания на остановках также учитывается. На любой остановке можно пересесть на любую маршрутку, которая прибывает на нее не раньше той, на которой Федор приехал на остановку (если две маршрутки приходят в одно и то же время, пересадка между ними тоже возможна).

Входные данные

В первой строке даны два числа, N и M ($2 \leq N \leq 10^5$, $1 \leq M \leq 10^5$) — количество остановок в городе и количество маршруток соответственно. Затем идет описание M маршруток.

Каждая маршрутка описывается следующим образом. Сначала задается число K_i — количество остановок, которые проезжает маршрутка, помимо остановки под номером один. Гарантируется, что сумма всех K_i не превышает 10^5 . Затем идут K_i строк, в каждой из которой по два натуральных числа, X_{ij} и T_{ij} ($2 \leq X_{ij} \leq N$, $1 \leq T_{ij} \leq 1000$) — номер остановки, в которую приезжает маршрутка, и время в секундах, необходимое, чтоб добраться в эту остановку из предыдущей (в обратную сторону маршрутке требуется такое же время). Гарантируется, что для каждой маршрутки ни одна остановка не повторяется дважды.

Выходные данные

Выведите единственное число — минимальное время, за которое Федор может добраться до Никиты или -1, если этого сделать невозможно.

Пример

Входные данные	Результат
4 7 1 2 6 1 4 8 3 3 4 2 20 4 11 1	3



Входные данные	Результат
3 11 1 2 6 1 4 3 3 3 17 2 19 4 10	
5 5 1 2 7 1 3 3 1 5 17 1 4 11 2 2 17 3 15	17

I. Фломастеры

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

У Никиты есть маленькая сестричка. Она очень любит рисовать. Однажды брат подарил ей набор разноцветных фломастеров, чтобы та рисовала и не мешала ему делать лабораторные работы.

Сестра у Никиты очень творческая, а творческие люди, насколько все мы знаем, любители беспорядка. Каждый раз после рисования очередного детского шедевра она разбрасывает открытые фломастеры и колпачки от них на столе. Убирать это всё приходится, конечно же, Никите.

И вот, в очередной раз в процессе уборки Никите показалось, что цвета колпачков и фломастеров немного не соответствуют (могли перемешаться с другим набором). Он не хочет в пустую тратить своё время и просит вас помочь определить, можно ли закрыть все фломастеры колпачками. Фломастер можно закрыть колпачком только если у них совпадает цвет.



Входные данные

Первая строка содержит число N ($1 \leq N \leq 10^4$). В следующих N строках находятся описания фломастеров, затем идут N строк, описывающие колпачки. Описание состоит из строки до 10 символов – названия цвета.

Выходные данные

Выведите «YES», если все фломастеры могут быть закрыты колпачками, иначе выведите «NO».

Пример

Входные данные	Результат
2 yellow blue blue yellow	YES
1 yellow black	NO



J. Binary Nuts

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

С наступлением осени в ореховой роще, где живет белка с бельчатами, появилось много орехов. Белка собирает их без усталости, приносит в дупло и снова отправляется собирать.

Чтобы бельчатам не скучно было оставаться одним, они придумали такую игру. Сначала они раскладывают собранные белкой орехи по одному для просушки. Потом соединяют их в кучки по 2 ореха и далее по такому принципу: если есть две кучки с одинаковым числом орехов (например, по два), то они объединяются в одну. После того, как все объединения завершились, подсчитывается число получившихся кучек и все орехи сбрасываются в дупло. Потом появляется белка, спрашивает, какое число получилось, оставляет новую порцию орехов и бельчата начинают новую игру.

Известно, что белка приносила орехи N раз в количестве $Q_1, Q_2, \dots, Q_{N-1}, Q_N$.

Входные данные

Первая строка содержит целое число N ($3 \leq N \leq 2000$) — количество порций принесенных орехов. В каждой из следующих N строк содержится число Q_i ($1 \leq Q_i \leq 10^9$) — количество орехов в i -ой порции.

Выходные данные

Нужно вывести для каждого запроса Q_i в отдельной строке одно целое число — количество получившихся для i -ой порции кучек.

Пример

Входные данные	Результат
3	2
5	1
16	3
7	



К. Виртуальная клавиатура

Ограничение по времени: для C/C++ — 4 секунды, для Java — 20 секунд

Ограничение по памяти: 256МБ

Входной / выходной файл: стандартный поток ввода / стандартный поток вывода

Как вы уже, наверное, поняли, Вася Пупкин очень любит RobotOS, и не в последнюю очередь — за её кастомизируемость. Например, в J-OS возможность установить стороннюю виртуальную клавиатуру появилась лишь недавно, а у Васи на его смартфоне с RobotOS клавиатура WsiftKey стоит уже много лет. Но вот несчастье, недавно WsiftKey продалась компании SoftMicro. Теперь, как сторонник операционной системы Lunix, Вася не может пользоваться WsiftKey и ему срочно нужна замена. Как настоящий программист, Вася решил написать свою собственную клавиатуру.

Одной из важнейших функций современной виртуальной клавиатуры, как вы знаете, являются подсказки слов. Проанализировав свои недавние переписки, Вася понял, что в основном общается с людьми одними и теми же фразами. Поэтому, на первое время модуль подсказок можно сделать очень простым. Достаточно, чтобы он запоминал введённые ранее предложения в словаре и выдавал в качестве подсказки окончание предложения, начинающегося с тех же слов, которые Вася сейчас печатает. Конечно, возможность удалить предложение из словаря тоже должна быть. Но, пока Вася занят реализацией поддержки вставки emoji и GIF анимаций (новая функция RobotOS 7.1), вам придётся помочь ему с этим модулем.

На вход модуля поступает три типа команд: добавить предложение в словарь, удалить предложение из словаря и вывести подсказку по заданным начальным словам. Команды пользователя проверяются в другом модуле, поэтому команд добавления предложений, которые уже есть в словаре, и удаления предложений, которых в нём нет, не будет.

Подсказки должны выдаваться следующим образом. Сначала нужно найти в словаре все предложения, которые начинаются с заданной последовательности слов (именно целых слов, а не символов). Если в словаре есть предложение, в точности совпадающее с заданной начальной последовательностью слов, его следует не учитывать при поиске. Затем следует выбрать лексикографически наименьшее из найденных предложений. В качестве подсказки следует выдать окончание этого предложения, то есть все слова, следующие за заданной начальной последовательностью. Если же подходящего предложения нет, в качестве подсказки нужно выдать смайлик, потому что Вася любит завершать предложения смайликами.

Входные данные

В первой строке указано количество команд C ($1 \leq C \leq 10^5$). Далее следует C строк, каждая из которых описывает одну команду.

Описание команды состоит из типа команды и последовательности слов. Тип команды задаётся одним символом: «+» значит добавить предложение в словарь, «-» — удалить, а «?» — выдать подсказку. За типом следует один пробел и последовательность слов, разделённых пробелами. Последовательность может содержать от 1 до 10 слов. Все слова состоят из не более 10 маленьких английских букв.

Выходные данные

Для каждой команды «?» выведите в отдельной строке подсказку согласно описанному выше алгоритму. При выводе окончания предложения из словаря пробел в самом начале строки выводить не следует. Если подходящей подсказки нет, выведите для соответствующего запроса последовательность символов «:-)» (без кавычек).



Пример

Входные данные	Результат
14 + hello world + hello world program + hello world app ? hello ? hello world - hello world app ? hello world - hello world program ? hello world + foo barrr baz ? foo ? foo bar ? foo barrr ? barrr	world app program :-) barrr baz :-) baz :-)

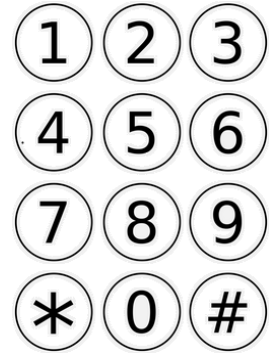
L. Телефонные номера

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ



Никита со своей девушкой гуляли по парку и случайно нашли на одной из лавочек у пруда мобильный телефон. Без долгих раздумий ребята решили вернуть телефон хозяину. Для этого они задумали позвонить на один из номеров в телефонной книжке и сообщить о том, что телефон у них. Но вот незадача – в телефоне огромное количество номеров. И, почему-то, ни одного имени – просто номера.



На какой же номер звонить? На случайный, сказали бы вы. Но ребята не любят случайные числа и хотят, чтобы их выбор был на чём-то основан. Выбор пал на самый оптимальный для набора номер. Номер должен быть таким, чтобы как можно меньше водить пальцами при наборе (движение пальца до самой первой кнопки не учитывается). Расстояние между кнопками равно $|x_1 - x_2| + |y_1 - y_2|$. Кнопки соответствуют стандартному мобильному циферблату, как на картинке. Например, кнопка 1 имеет координаты $(0, 0)$, а кнопка 0 – $(3, 1)$.

Если оптимальных номеров несколько, ребята выберут тот, который лексикографически меньший.

Помогите ребятам определить на какой номер звонить, так как у них мало времени – они еще хотят успеть погулять и покушать мороженое.

Входные данные

Первая строка содержит числа N ($1 \leq N \leq 10^4$) — количество номеров, и M ($1 \leq M \leq 20$) — количество цифр в номерах. Далее идут N строк длиной по M символов — номера телефонов. Все символы являются цифрами.

Выходные данные

Выведите номер, на который необходимо позвонить.

Пример

Входные данные	Результат
3 3 093 140 204	093
2 4 9241 1138	1138



М. Неправильный калькулятор

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Однажды, опоздав на пару, Вася Пупкин стал бродить по корпусу ФкАВТо (Факультет конечных Автоматов и Вычислений Тернарными операторами). Зайдя на этаж кафедры СПО (Самого Программного Обеспечения), он заметил на одной из скамеек для студентов калькулятор. На нём было фломастером написано «Включи меня!».

Вася включил калькулятор. Но вместо привычного «0» на дисплее появилось «35+7». «Интересно», — подумал Вася, набрал «42» и нажал равно. На секунду на дисплее калькулятора появилось «999999», а после этого — «154-31». «Какой-то неправильный калькулятор, — подумал Вася. — Наверное, над ним поработали любители кастомных прошивок». Вася набрал «123». На экране мигнуло «999998» и появилось «6×8». Вася ввёл «46». На экране мигнуло «1000000» и появился новый пример. «Чёрт, точно, 44 ведь!» — расстроился Вася. Тут ему стало интересно, что будет, если правильно решить миллион примеров подряд, ни разу не ошибившись.

Конечно, решать миллион примеров вручную — это слишком долго. К счастью, у Васи был с собой небольшой программируемый робот. Вася взялся программировать движения робота, а вас просит пока по-быстрее запрограммировать модуль принятия решений, поскольку пара, которую прогуливает Вася, не резиновая.

Встроенная нейросеть робота самостоятельно выполняет первичный анализ изображения с камер. На вход модуля принятия решений поступает картинка экрана калькулятора в виде ASCII графики. Каждый символ на экране изображён квадратом 5x5 ASCII символов. Соседние символы разделены одним столбцом пробелов. Известно, что символы, которые появляются на экране калькулятора, выглядят следующим образом:

1	2	3	4	5	6	7
8	9	0	+	-	*	/

Сверху вниз, слева направо: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, плюс, минус, умножить, разделить. Все символы занимают квадрат 5x5, на пустых местах стоят пробелы. В цифрах 1 и 4 и знаках действий часть строк и/или столбцов полностью заполнена пробелами.

Модуль принятия решений должен вычислить выражение и выдать ответ. Поскольку нейросеть робота имеет визуальное мышление, выход модуля принятия решений тоже должен быть в виде ASCII графики.

Входные данные

Входные данные состоят из ровно пяти строк, каждая из которых содержит одинаковое количество символов. Длина строк w удовлетворяет выражению $w=6s-1$ ($3 \leq s \leq 20$, $s \in \mathbb{Z}$). Строки могут содержать символы `-+/\|_X0` (минус, плюс, слэш, обратный слэш, вертикальная черта, подчёркивание, большая английская X, большая английская O), а так же



пробелы.

С помощью указанных символов нарисовано математическое выражение, которое необходимо вычислить. Оно имеет вид «левый операнд, оператор, правый операнд». Оба операнда — целые неотрицательные числа. Гарантируется, что формат ввода корректен, математическое выражение корректно и имеет смысл, а результат выражения — целое неотрицательное число.

Выходные данные

Выведите результат математического выражения в таком же формате, как входные данные (включая ограничение на максимальную длину строк). То есть, выведите пять строк одинаковой длины, содержащие указанные выше символы, которыми методом ASCII графики нарисован результат математического выражения. Не выводите лидирующие нули.

Нарушение формата вывода (неправильное количество или длина строк, посторонние символы, неправильная картинка, лидирующие нули) приведёт к вердикту Presentation Error. Неправильное числовое значение ответа приведёт к вердикту Wrong Answer.

Пример

Входные данные	Результат

Примечание

Вы можете скачать текстовый файл со всеми цифрами и примерами по ссылке: <https://acm.opu.ua/statements/onpu2017-for-M.txt>. Не копируйте из PDF файла условий, так как в нём может не хватать пробельных символов.



N. Числовой треугольник

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Рассмотрим числовой треугольник:

```
1
2+3
4+5+6
7+8+9+10
...
```

Строки этого треугольника нумеруются сверху вниз, начиная с единицы. Вычислите сумму, записанную выражением в заданной строке треугольника.

Входные данные

Входные данные содержат единственное целое число N ($1 \leq N \leq 10^6$) — номер строки числового треугольника, сумму чисел в которой нужно найти.

Выходные данные

Выведите единственное натуральное число — сумму чисел в заданной строке треугольника.

Пример

Входные данные	Результат
4	34