A. Отгадай слово

|  |  |
| --- | --- |
| Ограничение времени | 5 секунд |
| Ограничение памяти | 512Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Саша разрабатывает игру «Отгадай слово». В этой игре, игрок должен отгадать загаданное слово из N букв за несколько попыток.

В данный момент перед Сашей стоит задача написать логику проверки величины совпадения попытки игрока с загаданным словом.

Более формально, пусть есть строка S — загаданное слово и строка Q — попытка игрока. Обе строки имеют одинаковую длину N. Для каждой позиции 1≤i≤N строки Q, нужно вычислить тип совпадения в этой позиции со строкой S.

Если Qi=Si, то в позиции i тип совпадения должен быть равен correct.

Если Qi≠Si, но существует другая позиция 1≤j≤N, такая что Qi=Sj, то в позиции i тип совпадения должен быть равен present.

* Каждую букву строки S можно использовать не более чем в одном совпадении типа correct или present.
* Приоритет всегда отдается типу correct.
* Из всех возможных вариантов использования в типе present программа Саши выбирает самую левую позицию в строке Q.

В остальных позициях тип совпадения должен быть равен absent.

Формат ввода

В первой строке задана строка S (1≤∣∣S∣∣≤106) — загаданное слово.

Во второй строке задана строка Q (|Q|=|S|) — попытка игрока.

Гарантируется, что строки S и Q содержат только заглавные латинские буквы.

Формат вывода

Выведите N строк. В строке i должна находиться одна из строк correct, present или absent — результат совпадения в позиции i строки Q со строкой S.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| COVER  CLEAR | correct  absent  present  absent  correct |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| ABBA  AAAA | correct  absent  absent  correct |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| ABCBC  BBACA | present  correct  present  present  absent |

Примечания

Пояснение к первому тестовому примеру.

Q1=S1 и Q5=S5, поэтому для позиций 1 и 5 ответ correct.

Q3=S4, поэтому для позиции 3 ответ present.

Буквы L и A не встречаются в строке S, поэтому для позиций 2 и 4 ответ absent.

Пояснение ко второму тестовому примеру.

Q1=S1 и Q4=S4, поэтому для позиций 1 и 4 ответ correct.

Буква A больше не встречается в строке S, поэтому для позиций 2 и 3 ответ absent.

Пояснение к третьему тестовому примеру.

Q2=S2, поэтому для позиции 2 ответ correct.

Q1=S4, Q3=S1 и Q4=S3, поэтому для позиции 1, 3 и 4 ответ present.

Буква A встречается в строке S в позиции 1, но она уже участвует в совпадении present (Q3=S1), поэтому для позиции 5 ответ absent.

B. Поздравляю, вы прошли

|  |  |
| --- | --- |
| Ограничение времени | 3 секунды |
| Ограничение памяти | 512Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Известная IT-компания Тындекс собирается открыть новый офис на Марсе. Так как подготовка к полету займет достаточно большое время, отбор кандидатов на вакансии было решено начать уже сейчас.

Так как желающих оказалось очень много, первая часть отбора прошла в виде соревнования по решению задач — на каждую вакансию было предложено 6 задач одинаковой ценности. Приоритет отдаётся кандидатам, решившим больше задач.

При равенстве количества решенных задач кандидаты сравниваются по целочисленному штрафу - — чем он меньше, тем выше приоритет у кандидата. Штраф определяется по особой формуле, что гарантирует отсутствие двух кандидатов с одинаковым количеством задач и одинаковым штрафом.

Так как отбор в марсианский офис проходит «чуть сложнее», чем в любой земной, для каждой вакансии определено максимальное количество приглашенных кандидатов — больше данной величины приглашать нельзя ни при каком условии.

На вас возложена очень важная задача — по информации о вакансиях и о результатах отборочного соревнования вывести всех кандидатов, прошедших в следующую часть отбора.

Формат ввода

В первой строке содержится целое число n (1≤n≤104) — число открытых вакансий в новом офисе.

Следующие n строк имеют вид si,mi (1≤|si|≤30, 1≤mi≤104) — название и максимальное число кандидатов на i-ю вакансию. Название вакансии содержит только строчные латинские буквы и знак подчеркивания ‘\_’.

Далее следует строка, содержащая целое число k (1≤k≤105) — число участвовавших в отборочном соревновании кандидатов.

Последующие k строк имеют вид cj,qj,rj,pj (1≤∣∣cj∣∣≤30, 0≤rj≤6, 0≤pj≤109) — строковый идентификатор j-го кандидата, название интересующей его вакансии, количество решенных кандидатом задач и начисленный ему штраф соответственно. Идентификатор кандидата содержит только строчные латинские буквы и знак подчеркивания ‘\_’.

Гарантируется, что:

* Интересующие кандидатов вакансии qj обязательно представлены во входных данных;
* Названия всех вакансий si попарно различны между собой;
* Идентификаторы кандидатов cj попарно различны между собой;
* Не существует двух кандидатов a и b, что ra=rb и pa=pb (или различаются количества решенных задач, или различается начисленный кандидатам штраф).

Все строки используют разделитель ‘,’ (запятая).

Формат вывода

Выведите список идентификаторов всех кандидатов, прошедших в следующий этап отбора на интересующие их вакансии в лексикографическом (алфавитном) порядке.

Кандидат считается прошедшим по вакансии i, если существует строго менее mi кандидатов на i-ю вакансию с большим, чем у данного кандидата, приоритетом (см. условие про сравнение кандидатов).

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2  ceo,1  co\_founder,1  3  arcady\_volozh,ceo,6,100  elon\_musk,ceo,5,0  ilya\_segalovich,co\_founder,6,10 | arcady\_volozh  ilya\_segalovich |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 1  co\_founder,1  2  elon\_musk,co\_founder,6,200  ilya\_segalovich,co\_founder,6,100 | ilya\_segalovich |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2  developer,2  hacker,3  5  anonymous,hacker,6,0  bjarne\_stroustrup,developer,6,1  julian\_assange,hacker,5,100500  bill\_gates,developer,3,1  guccifer,hacker,2,0 | anonymous  bill\_gates  bjarne\_stroustrup  guccifer  julian\_assange |

Пример 4

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2  plant,2  gardener,1  5  demeter,gardener,4,12  acacia,plant,0,5  cactus,plant,0,1  ficus,plant,0,4  palm,plant,0,3 | cactus  demeter  palm |

Примечания

В первом тестовом примере на вакансию «ceo» претендуют два кандидата — у ‘arcady\_volozh’ решено 6 задач и 100 штрафа, у ‘elon\_musk’ — решено 5 задач и 0 штрафа. В первую очередь сравнение идет по задачам: 6>5, поэтому в следующий этап проходит именно ‘arcady\_volozh’.

Во втором тестовом примере у обоих претендентов одинаковое количество решенных задач, но у ‘ilya\_segalovich’ штраф 100, что меньше штрафа 200 у ‘elon\_musk’. Поэтому в следующий этап проходит ‘ilya\_segalovich’.

В третьем тестовом примере важно отметить, что в ответе прошедшие кандидаты идут в объединенном списке отсортированные в лексикографическом порядке независимо от своего приоритета при отборе и выбранной ими вакансии.

В четвертом тестовом примере развернулась борьба за должность офисного растения. Хотя все кандидаты решили 0 задач на соревновании, ‘cactus’ и ‘palm’ набрали меньше всего штрафа, поэтому именно их приглашают в следующий этап отбора.

C. Отчет по товарам

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 512Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Недавно в Выньдекс.рынке среди покупателей провели опрос — какие товары они считают наиболее «интересными» для себя. На выбор предлагалось 5 различных вариантов фильтра:

* «Наименование товара содержит подстроку в любом регистре» (внутренний ключ ‘NAME\_CONTAINS’);
* «Цена больше или равна чем» (внутренний ключ ‘PRICE\_GREATER\_THAN’);
* «Цена меньше или равна чем» (внутренний ключ ‘PRICE\_LESS\_THAN’);
* «Товар поступил в продажу не позднее» (внутренний ключ ‘DATE\_BEFORE’);
* «Товар поступил в продажу не ранее» (внутренний ключ ‘DATE\_AFTER’);

По итогам опроса определились самые актуальные значения каждого из фильтров (по одному значению на фильтр).

Вам, как аналитику Выньдекс.Рынка, поставили задачу из имеющегося списка товаров выбрать все товары, удовлетворяющие актуальным значениям всех указанных фильтров.

Формат ввода

Общее описание формата входных данных:

Первая строка входных данных содержит список товаров в формате JSON.

Следующие 5 строк имеют вид qivi — фильтр и соответствующее ему актуальное значение.

Подробное описание формата списка товаров

Гарантии по формату JSON:

* нет запятых после последнего элемента массива;
* все имена полей и строки обернуты в двойные кавычки.

Обозначим количество товаров в списке через N. Гарантируется, что 0≤N≤1000.

Каждый товар в списке содержит следующую информацию (порядок полей не является фиксированным):

* целое число id (0≤id≤231−1) — уникальный идентификатор. Гарантируется, что идентификаторы всех товаров попарно различны;
* строка name (1≤|name|≤100) — наименование. Гарантируется, что наименование содержит только строчные и заглавные латинские буквы, а так же пробел;
* целое число price (0≤price≤231−1) — цена;
* строка date в формате «dd.MM.yyyy» (01.01.1970≤date≤31.12.2070) — дата поступления в продажу.

Подробное описание формата фильтров

Гарантируется, что:

* все qi различны между собой;
* qi является строкой из множества (NAME\_CONTAINS, PRICE\_GREATER\_THAN, PRICE\_LESS\_THAN, DATE\_BEFORE, DATE\_AFTER);
* в фильтре ‘NAME\_CONTAINS’ vi представляет из себя строку (1≤|vi|≤100), содержащую только строчные и заглавные латинские буквы;
* в фильтрах ‘PRICE\_GREATER\_THAN’ и ‘PRICE\_LESS\_THAN’ vi представляет из себя целое число (0≤vi≤231−1);
* в фильтрах ‘DATE\_BEFORE’ и ‘DATE\_AFTER’ vi представляет из себя строку в формате «dd.MM.yyyy» (01.01.1970≤vi≤31.12.2070).

Формат вывода

Выведите в формате JSON список товаров, удовлетворяющих всем указанным во входных данных фильтрам. Каждый товар должен быть выведен ровно один раз в отсортированном по возрастанию id порядке.

Выводить JSON допустимо как c дополнительными отступами и переводами строк, так и в одну строку.

Имена полей необходимо выводить в двойных кавычках.

Допустимо выводить запятую после последнего поля объекта или последнего элемента массива.

Каждый товар должен содержать информацию, аналогичную информации из входных данных:

* целое число id — уникальный идентификатор;
* строка name — наименование;
* целое число price — цена;
* строка date в формате «dd.MM.yyyy» — дата поступления в продажу.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| [{"id": 1, "name": "Asus notebook","price": 1564,"date": "23.09.2021"},{"id": 2, "name": "Earpods", "price": 2200, "date": "10.01.2022"},{"id": 3, "name": "Keyboard", "price": 2500, "date": "05.06.2020"}, {"id": 4, "name": "Dell notebook","price": 2300,"date": "23.09.2021"}]  NAME\_CONTAINS notebook  PRICE\_GREATER\_THAN 2000  PRICE\_LESS\_THAN 2400  DATE\_AFTER 12.09.2021  DATE\_BEFORE 02.01.2022 | [{"id": 4, "name": "Dell notebook", "price": 2300, "date": "23.09.2021"}] |

Примечания

При написании решения на Java можно выбрать комплятор «Java 8 + json-simple». В этом случае вы сможете воспользоваться библиотекой [json-simple](https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple/1.1.1) для парсинга и сериализации JSON

D. Ориентация в лабиринте

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 512Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Януш Воронов решил провести очередную выставку своих работ. В качестве места проведения он выбрал один из этажей заброшенного здания прямоугольной формы N×M метров.

У Януша есть схема этажа в виде N×M клеток (каждая клетка задаёт пространство площадью 1×1 метров), где «#» обозначает кусок стены, а «.» — пространство, доступное для перемещения посетителей.

Также на карте ровно одна клетка обозначена как «S» — участок, из которого посетители начнут осмотр выставки.

Гарантируется, что планировка этажа удовлетворяет следующим условиям:

* Все клетки в первых и последних строках / столбцах схемы являются стенами.
* От стартовой клетки можно добраться до любой пустой клетки, перемещаясь только вверх / вниз / влево / вправо.
* Между любой парой пустых клеток на схеме существует ровно один путь, возможно проходящий через стартовую клетку.

Януш хочет добиться идеального впечатления от осмотра выставки, поэтому хочет учесть направление, в котором посетитель впервые зайдет в каждую свободную клетку. Гарантируется, что такое направление определяется однозначно.

Помогите Янушу и выведите для каждой клетки направление, в котором посетитель впервые зайдет в эту клетку при осмотре выставки.

Формат ввода

В первой строке даны два целых числа N и M (3≤N,M≤500) — количество строк и столбцов на схеме этажа.

В следующих N строках расположено по M символов из множества (#,.,S).

Гарантируется, что

* Все клетки в первых и последних строках / столбцах схемы равны #.
* На схеме расположена ровно одна стартовая клетка S.
* От стартовой клетки можно добраться до любой пустой клетки, перемещаясь только вверх / вниз / влево / вправо.
* Между любой парой пустых клеток на схеме существует ровно один путь, возможно проходящий через стартовую клетку.

Формат вывода

Выведите N строк по M символов в каждой — схему этажа, где каждая пустая клетка . заменена на направление первого захода в эту клетку.

Занумеруем все строки от 1 до N сверху вниз, все столбцы — от 1 до M слева направо. В таком случае пустая клетка (r,c) должна содержать:

* L — если в клетку (r,c) зашли из клетки (r,c+1);
* R — если в клетку (r,c) зашли из клетки (r,c−1);
* U — если в клетку (r,c) зашли из клетки (r+1,c);
* D — если в клетку (r,c) зашли из клетки (r−1,c).

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5 8  ########  #......#  #.#S#.##  ##...###  ######## | ########  #LLURRR#  #D#S#D##  ##LDR###  ######## |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3 3  ###  #S#  ### | ###  #S#  ### |

Примечания

Рассмотрим первый тестовый пример.

Стартовой является клетка (3,4).

Из стартовой клетки посетитель может попасть в клетку (2,4), сделав шаг вверх, или в клетку (4,4), пройдя вниз.

Из клетки (2,4) можно пойти налево в клетки (2,3) и (2,2); из клетки (2,2) можно дойти до клетки (3,2), пройдя вниз.

Также из клетки (2,4) можно пройти направо в клетки (2,5), (2,6) и (2,7); из клетки (2,6) можно попасть в клетку (3,6), пройдя вниз.

Из клетки (4,4) можно пройти всего в две клетки — налево в (4,3) и направо в (4,5).

Во втором тестовом примере посетитель никуда не может пройти из стартовой клетки.

# E. В одном шаге от идеала

|  | Все языки | Python 3.7 (PyPy 7.3.3) | Python 3.10.1 |
| --- | --- | --- | --- |
| Ограничение времени | 6 секунд | 10 секунд | 15 секунд |
| Ограничение памяти | 64Mb | 64Mb | 64Mb |
| Ввод | input.txt | | |
| Вывод | стандартный вывод или output.txt | | |

Василиса недавно узнала про «правильные скобочные последовательности» — последовательности из «(» и «)» такие, что существует хотя бы одно разбиение всех скобок последовательности на пары «(» и «)», для которых верно:

* в любой паре порядковый номер «(» в последовательности меньше, чем у соответствующей ей «)».
* каждая скобка относится ровно к одной паре из разбиения.

Например, последовательность «(()())()» правильная, так как существует разбиение (1,6), (2,3), (4,5), (7,8) (скобки нумеруются с 1 слева направо).

Примеры неправильных скобочных последовательностей:

* «((()» неправильная, так как в единственно возможном разбиении (1,4), (2,4), (3,4) скобка 4 используется во всех трёх парах.
* «)(» неправильная, так как в единственно возможном разбиении (2,1) позиция скобки «(» больше, чем позиция «)».
* «((» неправильная, так как невозможно построить ни одной пары без скобок «)».

Василисе очень понравилась данная тематика, поэтому ей захотелось привнести туда чего-то нового и необычного. Поэтому Василиса придумала «почти правильную скобочную последовательность» — скобочную последовательность, из которой можно получить правильную скобочную последовательность удалением ровно одной скобки.

К примеру, последовательность «(()» является почти правильной, так как удаление любой скобки «(» приведет к правильной последовательности «()».

Последовательность «)» тоже является почти правильной, так как удаление единственной скобки приведет к пустой последовательности скобок, которая является правильной по определению.

У Василисы есть файлы с её решениями домашних заданий по математике. Она решила для каждого задания выяснить, является ли последовательность скобок в этих выражениях почти правильной — и, если это так, то найти наименьшую позицию скобки, после удаления которой последовательность скобок станет правильной.

Обратите внимание, что Василису не интересует корректность получающегося после удаления скобки выражения — она всецело сосредоточена только на корректности скобочной последовательности.

Так как у Василисы очень большие решения, то ей одной не справиться — помогите ей в этом нелегком деле.

## Формат ввода

Обратите внимание, что в данной задаче разрешен только ввод из файла «input.txt».

В единственной строке файла содержится строка S (1≤∣∣S∣∣≤5⋅107) — решение одного из заданий по математике. Строка гарантированно завершается символом перевода строки.

Гарантируется, что строка S содержит символы только из указанных ниже:

* строчные английские буквы a - z;
* знаки математических операций: +, −, ∗, ∕, =;
* скобки ( и );
* пробел.

## Формат вывода

Если последовательность скобок в решении является почти правильной, то выведите наименьшую позицию символа в строке, удаление которого сделает последовательность скобок правильной. В ином случае выведите число −1.

### Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| a + b = b + a | -1 |

### Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| d + (a + (b + c) = (a + b) + c + d | 5 |

### Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| (a((b + c) = ab + bc | -1 |

## Примечания

Внимание! При отправке решений на языке Python настоятельно рекомендуется использовать компилятор PyPy 7.3 вместо Python 3.10.

В первом тестовом примере последовательность скобок пустая, а значит она уже правильная, ничего удалять не нужно — поэтому ответом является −1.

Во втором тестовом примере последовательность скобок «(()()» является почти правильной. Чтобы сделать её правильной, необходимо удалить одну из двух открывающих скобок в начале скобочной последовательности (позиции в строке 5 и 10). Так как требуется удалить символ на наименьшей позиции, то ответом является 5.

В третьем тестовом примере последовательность скобок «((()» не является почти правильной, так как удалением ровно одной скобки невозможно получить правильную последовательность — поэтому ответ −1.