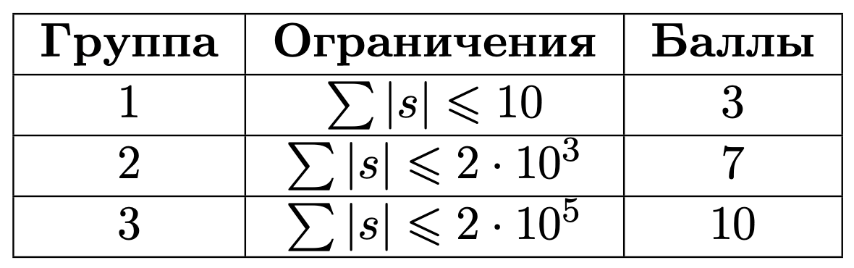
1. **Вставка символов**

**Условие задачи**

Изначально у Игоря была строка, состоящая из одинаковых символов. Над ней производилось некоторое количество операций вставки символа. Очередной символ можно вставить только между любыми двумя одинаковыми символами. Вам дана строка s, состоящая из латинских символов как нижнего, так и верхнего регистра. Определите, могла ли эта строка получиться после некоторого (возможно нулевого) количества таких операций над ней.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число t — количество наборов входных данных, (1≤t≤105). Каждый набор входных данных состоит из единственной строки s, (1≤∣s∣≤2⋅105).



**Выходные данные**

Для каждого набора входных данных выведите YES, если строка *s* могла получиться после описанных операций, иначе выведите NO. В первом примере изначально строка имела вид *aaaa*. Далее вставили символы *b* и *c* после первого и второго символа *a* соответственно.

|  |  |
| --- | --- |
| **Пример теста 1**  **Входные данные**  2  abacaa  PppP  **Выходные данные**  YES  NO | **Пример теста 2**  **Входные данные**  2  X  kkk  **Выходные данные**  YES  YES |

1. **Три банка, три валюты**

**Условие задачи**

Банки A, B и C предлагают обмен валюты. Каждый банк меняет рубли, доллары и евро по своему курсу. В каждый банк можно обратиться только 1 раз. Одно обращение — это одна операция обмена валюты. В одном банке нельзя совершить несколько операций обмена валют. Вам нужно обменять 1 рубль в максимально возможное количество долларов.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число *t* (1≤t≤10000) — количество наборов входных данных. Далее следуют описания наборов входных данных. Каждый набор входных данных содержит описания обменных курсов трех банков. Описание обменного курса одного банка состоит из шести строк, каждая из которых содержит по два целых числа *n* (1≤n≤100) и *m* (1≤*m*≤100) — курс обмена одной валюты на другую: • Курс обмена рублей на доллары. • Курс обмена рублей на евро. • Курс обмена долларов на рубли. • Курс обмена долларов на евро. • Курс обмена евро на рубли. • Курс обмена евро на доллары.

**Выходные данные**

Для каждого набора входных данных в отдельной строке выведите одно число — максимальное количество долларов, которое можно получить, если совершать не больше одного обмена в банке. Ответ будет считаться правильным, если его относительная или абсолютная погрешность от верного не превосходит 10−610−6.

**Пример теста 1**

**Входные данные**

1

100 1

100 1

1 100

3 2

1 100

2 3

100 1

100 1

1 100

3 2

1 100

2 3

100 1

100 1

1 100

3 2

1 100

2 3

**Выходные данные**

0.015

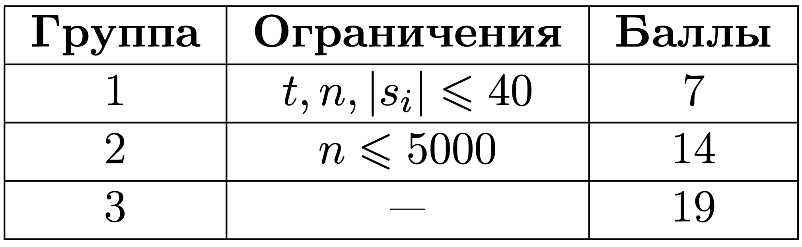
1. **Похожие строки**

**Условие задачи**

Маркетологам компании «Ozion»‎ поступило задание — придумать уникальный слоган для новой рекламной компании. У компании очень большая и креативная команда, поэтому слоган решили собирать из слов. Для этого каждый сотрудник должен предложить какое-то слово для слогана. Вы работает программистом в компании «Ozion»‎ и хотите помочь. Маркетологам интересно, как много похожих слов они вам прислали. Вам дается число *t* — общее количество наборов со строками. Каждый набор содержит *N* строк. Назовём 2 строки похожими, если у них совпадают все буквы на чётных позициях или все буквы на нечётных. Ваша задача — помочь креативным маркетологам найти количество пар похожих строк.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Первая строка каждого набора входных данных содержит целое число *t* — количество тестовых наборов, (1≤t≤100). Вторая строка каждого набора входных данных содержит целое число *n* — количество строк, (1≤n≤106). Следующие *n* строк каждого набора входных данных содержат множество строк *si*​ (1≤∣si∣≤106). Строки состоят из строчных латинских букв. Гарантируется, что сумма *n* по всем наборам входных данных не больше 106. Гарантируется, что сумма длин всех строк по всем наборам входных данных не больше 106.



**Выходные данные**

Для каждого набора входных данных выведите количество пар похожих строк. Похожие пары: *abcde* и *adcbe*, *abcde* и *xbxde*.

**Пример теста 1**

**Входные данные**

7\n 3\n ababa\n ababa\n ababa\n 3\n asd\n das\n sda\n 2\n abca\n abc\n 4\n aaaa\n aaaa\n aaaa\n aaa\n 2\n aa\n aa\n 2\n a\n a\n 2\n a\n b\n

**Выходные данные**

3

0

1

6

1

1

0

1. **Крестики-нолики (Junior)**

**Условие задачи**

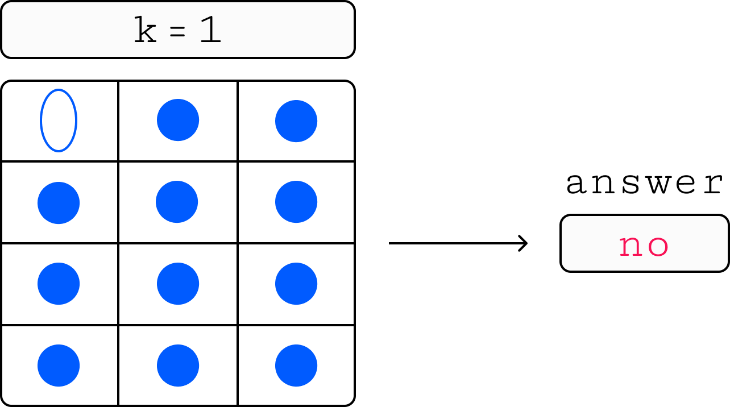
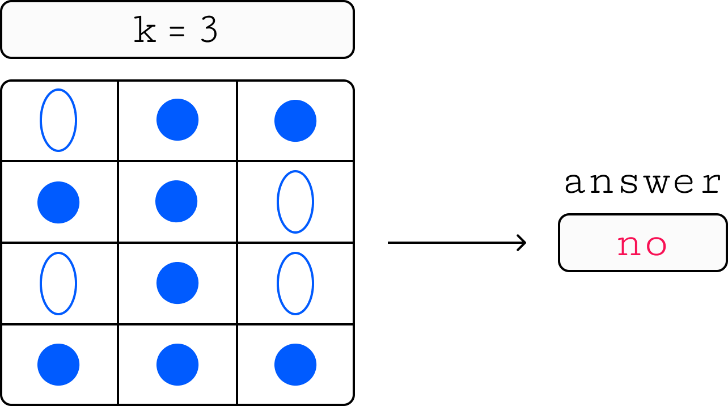
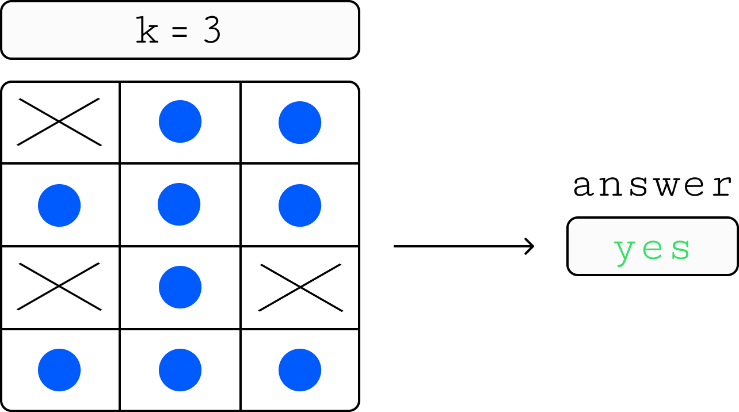
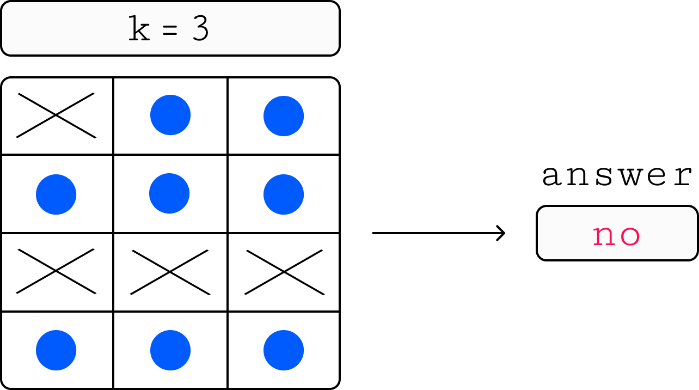
Перед вами **расширенная** версия игры крестики-нолики. В привычной всем игре победившим считается тот, кто собрал первым последовательность из трёх крестиков или ноликов в одной строке, столбце или диагонали на доске из трёх строк и трёх столбцов. В расширенной версии игры победителем считается тот, кто собрал первым последовательность из *k* крестиков или ноликов на доске из *n* строк и *m* столбцов. Ваша задача — по доске с некоторым (не обязательно корректным) состоянием игры понять, можно ли поставить ровно один крестик так, чтобы крестики победили. Например, если в данном состоянии доски уже есть последовательность из хотя бы *k* крестиков или ноликов, то победитель уже есть, а значит, ровно один крестик для победы поставить нельзя.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число *t* (1≤t≤103) — количество наборов входных данных. Далее следуют описания наборов входных данных. Первая строка каждого набора входных данных содержит одно целое число *k* (1≤k≤105) — необходимая для победы длина строки, столбца или диагонали. Вторая строка каждого набора входных данных содержит два целых числа *n* и *m* (1≤n≤104,1≤m≤107) — количество строк и столбцов на доске. Следующие *n* строк каждого набора входных данных содержат по *m* символов в каждом, где X обозначает крестик, 0 обозначает нолик, а **.** обозначает пустую клетку. Гарантируется что суммарный размер досок по всем наборам входных данных не превосходит 107, то есть ∑n⋅m≤107. 

**Выходные данные**

Для каждого набора входных данных выведите YES, если можно поставить ровно один крестик для победы, иначе выведите NO.



**Пример теста 1**

**Входные данные**

3\n 3\n 3\n 3\n X..\n ..O\n OOX\n 2\n 5 3\n ...\n O.O\n X.O\n ...\n ...\n 3\n 5 5 X.X..\n .....\n .OX..\n ..O..\n ...O.\n

**Выходные данные**

YES

NO

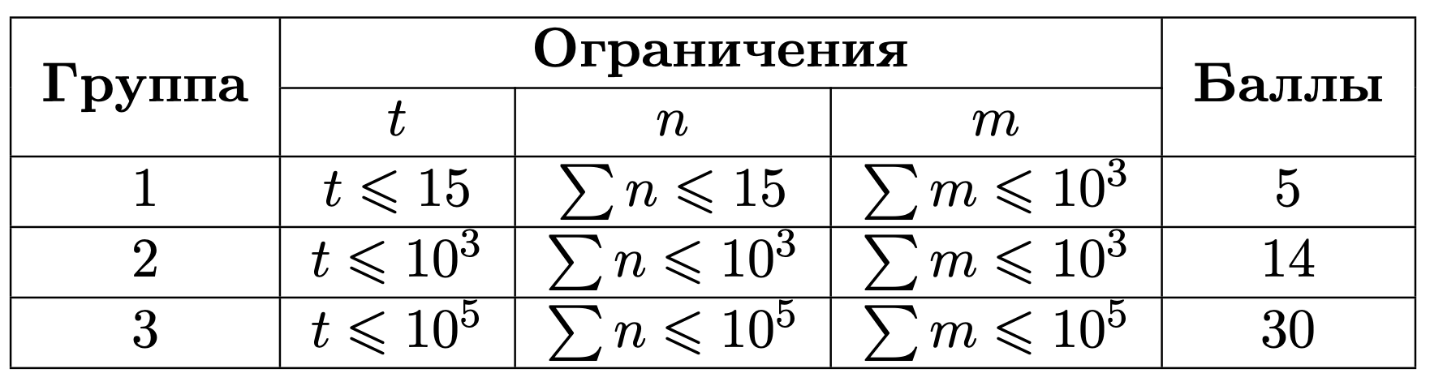
NO

1. **Галерея искусств (Junior)**

**Условие задачи**

В королевстве Артистия готовится грандиозная выставка, которая пройдет в знаменитой Галерее искусств. Мастера со всего королевства прислали свои лучшие картины, и кураторам выставки предстоит сложная задача — разместить все произведения искусства в специальные защитные коробки, чтобы транспортировать их в галерею. У вас есть *n* коробок разного размера, каждая с определенной шириной *ai*​ и длиной *bi*​. Также у вас есть *m* картин, каждая с заданными шириной *cj*​ и длиной *dj*​. Каждая картина должна быть упакована в коробку, в которую она полностью вмещается (минимальная и максимальная стороны картины должны быть меньше либо равны минимальной и максимальной сторонам коробки соответственно), при этом коробки могут уместить неограниченное количество картин. Вам необходимо найти способ упаковать все картины, используя минимальное количество коробок, чтобы все картины благополучно достигли Галереи искусств.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число *t* (1≤t≤105) — количество наборов входных данных. Далее следует описание наборов входных данных. Первая строка каждого набора входных данных содержит одно целое число *n* (1≤n≤105) — количество коробок. В следующих *n* строках каждого набора входных данных содержатся два целых числа *ai*, bi (1≤ai,bi≤109) — ширина и длина коробки *i*. В следующей строке каждого набора входных данных содержатся одно целое число *m* (1≤m≤105) — количество картин. В следующих *m* строках каждого набора входных данных содержатся два целых числа cj, dj (1≤cj,dj≤109) — ширина и длина картины *j*. 

**Выходные данные**

Выведите одно целое число — минимальное количество коробок, необходимых для упаковки всех картин. Если ответа нет — выведите −1−1. Для первого набора входных данных чтобы упаковать все коробки нам необходимы коробки под номером 1 и 3: .. в 1-й коробке помещается 2-я картина (1≤11≤1 и 9≤109≤10); .. в 3-й коробке помещается 1-я (2≤22≤2 и 4≤54≤5) и 3-я (2≤22≤2 и 2≤52≤5) картины. Для второго набора входных картину нельзя поместить ни в одну из коробок.

**Пример теста 1**

**Входные данные**

2\n 3\n 1 10\n 3 3\n 5 2\n 3\n 4 2\n 9 1\n 2 2\n 2\n 1 10\n 3 3\n 1\n 4 2

**Выходные данные**

2

-1

1. **Content delivery**

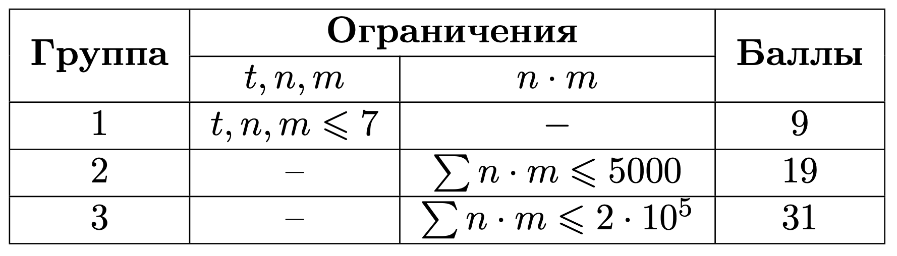
**Условие задачи**

В дата-центре крупной компании «‎Zono»‎ находится *n* серверов. Каждый сервер *i* обладает собственной пропускной способностью, обозначенной как *throughputi*​ в единицах: память/секунду. Также есть *m* изображений, каждое из которых занимает *weightj*​ единиц памяти. Каждый сервер способен доставлять изображения клиентам. Время доставки изображения зависит от его размера и пропускной способности сервера. Время доставки изображения j*j* сервером *i* вычисляется как *timei*,*j*​=*ceil*(*weightj*​/*throughputi*​), где ceil(x)*ceil*(*x*) — округление в большую сторону до целого числа.

Пример: Для сервера с пропускной способностью throughput=6, доставляющего изображения с размерами [12,15,18,4,32], время доставки составит [2,3,3,1,6] соответственно. Ваша задача — определить для каждого изображения, на каком сервере оно должно храниться, чтобы минимизировать разницу во времени между самым быстрым и самым медленным временем доставки, среди всех изображений. Чем меньше эта разница, тем стабильнее и равномернее будет происходить доставка.

**Входные данные**

Каждый тест состоит из нескольких наборов входных данных. Далее следует описание наборов входных данных. Первая строка содержит одно целое число *t* (1≤t≤2⋅105) — количество наборов входных данных. Первая строка каждого набора входных данных содержит одно целое число *n* (1≤*n*≤2⋅105) — количество серверов. Вторая строка каждого набора входных данных содержит n*n* целых чисел *throuputi*​ (1≤*throuputi*​≤109) — пропускная способоность *i*-го сервера. Третья строка каждого набора входных данных содержит одно целое число *m* (1≤*m*≤2⋅105) — количество изображений. Четвёртая строка каждого набора входных данных содержит *m* целых чисел weightj (1≤*weightj*​≤109) — размер *j*-го изображения. Гарантируется, что сумма *n*∗*m* по всем наборам входных данных не больше 2⋅105.



**Выходные данные**

В первой строке выведите минимальную разность между самым быстрым и самым медленным временем доставки. Во второй строке для каждого изображения выведите номер сервера, на котором оно должно храниться, чтобы достичь этой разности. Если имеется несколько возможных вариантов распределения изображений по серверам, при которых достигается минимальная разность — выведите любой из вариантов. Если в первом наборе выходных данных 1 и 2 изображение доставляется вторым сервером, а 3 и 4 — первым, то время доставки будет соответственно [12/5,14/5,7/3,9/3]=[3,3,3,3]. Разность будет равна 0. Если во втором наборе входных данных 1 изображение доставляется первым сервером, а 2, 3, 4, 5 — вторым, то время доставки будет соответственно [12/3,13/5,14/5,15/5,16/5]=[4,3,3,3,4]. Разность будет равна 11. В этом случае есть несколько возможных вариантов распределения изображений по серверам, при которых достигается минимальная разность. Можно показать что для этих тестов нельзя найти меньшую разность.

**Пример теста 1**

**Входные данные**

2

2

3 5

4

12 14 7 9

2

3 5

5

12 13 14 15 16

**Выходные данные**

0

2 2 1 1

1

1 2 2 2 2