

Задача А. Копипаста

При подготовке к Московской олимпиаде по информатике Вася решал задачи прошлых лет. В одной из задач необходимо было вывести N букв "А". Для решения задачи Вася решил воспользоваться текстовым редактором. Он набрал одну букву "А" и в дальнейшем пользовался одной из двух операций:

1. Скопировать весь напечатанный текст в буфер обмена
2. Вставить содержимое буфера обмена в конец файла

Последний скопированный буфер можно вставлять несколько раз. Определите, какое минимальное количество таких операций требуется Васе для получения N букв "А".

В первой строке вводится число t — количество тестов. Каждый тест описывается одним числом — количеством букв "А", которые необходимо получить.

Для каждого теста выведите минимальное количество операций.

Оценка за каждый тест — 10 баллов. Баллы начисляются за точное определение минимального количества операций.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. Во время тура проверяется, что решение содержит t чисел. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

стандартный ввод	стандартный вывод
3 6 8 4	5 6 4

Задача В. Агроастрономия

Вася проводит исследование, посвященное влиянию активности звезд на урожай редиски.

Каждому сорту редиски он сопоставил одну звезду. Сорт редиски характеризуется числом D — количеством дней необходимых от посадки до созревания.

Каждая звезда характеризуется интенсивностью излучения, выраженной целым числом. День называется благоприятным для посадки редиски, если излучение звезды было стабильным, т.е. в течение D дней от посадки до созревания, включая день посадки, разница в максимальном и минимальном уровне излучения звезды не превосходила числа I .

По известным уровням излучения и числам D и I определите все благоприятные дни для посадки редиски.

В первой строке вводится число t — количество сортов редиски и соответствующих им звезд.

Описание сорта редиски состоит из двух строк: первая содержит числа N , D и I , во второй содержится N чисел, задающих уровень излучения звезды.

Для каждого сорта выведите количество благоприятных для посадки редиски дней, а затем номера этих дней в порядке возрастания. Нумерация дней начинается с единицы.

Оценка за определение благоприятных дней для каждого сорта — 10 баллов. Баллы начисляются только в случае, если количество дней и сами благоприятные дни определены верно.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. Во время тура проверяется соответствие формату вывода (количество номеров дней совпадает с количеством указанным участником). Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

стандартный ввод	стандартный вывод
2	3
7 3 0	1 4 5
1 1 1 2 2 2 2	2
4 2 1	1 3
1 2 4 4	

Задача С. Разбиение на команды

Тренер Игорь очень любит готовить школьников к командным олимпиадам по программированию.

Всего к нему в кружок ходит N (N делится на 3) школьников, и, так как Игорь уже довольно опытный тренер, то для каждой пары школьников a и b он знает «коэффициент сыгранности» $C_{a,b}$ этих двух школьников. По его наблюдениям, если в одной команде находятся ученики под номерами a, b, c , успех их команды можно будет выразить как $C_{a,b} \cdot C_{b,c} \cdot C_{a,c}$. Игорю будет приятно, если как можно больше школьников успешно выступят на предстоящей Межгалактической Командной Олимпиаде Школьников по Программированию (МКОШП), и он хочет максимизировать суммарный успех составленных команд.

Иными словами, Игорь хочет максимизировать сумму $C_{a_i,b_i} \cdot C_{b_i,c_i} \cdot C_{a_i,c_i}$ по всем i от 1 до $\frac{N}{3}$, где a_i, b_i, c_i — номера учеников i -той команде. Каждый ученик должен быть распределен в какую-либо команду и только в одну.

Его кружок очень популярен, и он не справляется с тем, чтобы оптимально решить эту задачу, так помогите же ему!

В первой строке вводится количество учеников N ($1 \leq N \leq 675$, N делится на 3).

В последующих N строках вводятся по N чисел: в j -ом столбце i -ой строки находится коэффициент сыгранности учеников под номерами i и j .

Выведите оптимальное разбиение на команды в виде $\frac{N}{3}$ строк, в каждой из которых находится по 3 целых числа — номера учеников в этой команде. Ученики нумеруются с единицы

Оценкой за решение одного набора входных данных будет величина $10 \cdot \left(\frac{\text{part_solution}}{\text{jury_solution}}\right)^3$, где jury_solution — это лучшее решение среди всех участников и решения жюри, а part_solution — решение участника.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. Во время тура проверяется, что каждый участник распределен в одну из команд. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

стандартный ввод	стандартный вывод
1	1 6 4
6	2 3 5
0 1 1 1 1 2	
1 0 3 1 1 1	
1 3 0 1 1 1	
1 1 1 0 1 1	
1 1 1 1 0 1	
2 1 1 1 1 0	

Задача D. Нарезка текста

Сначала напомним, что расстояние Левенштейна между двумя строками s и t — это минимальное число операций: «добавить символ», «удалить символ» и «изменить один символ на другой», которое нужно применить к строке s , чтобы получить строку t . Например, расстояние между строками $s = aba$ и $t = dac$ равняется 3: можно применить такую последовательность операций: $aba \rightarrow ab \rightarrow ac \rightarrow dac$.

Длинный текст s был зачитан, записан и разрезан на несколько кусков. Затем каждый из кусков был распознан и преобразован обратно в текст — получился набор строк A . К сожалению, преобразования были неидеальными. В результате в текст могли добавиться или пропасть слова, также некоторые слова могли измениться на похожие.

Строка s и каждая строка из A имеет такие свойства: каждый символ в этих строках — либо маленькая латинская буква, либо пробел. Никакая строка не может начинаться или заканчиваться на пробел, в строках не может быть двух пробелов подряд.

Ваша задача: разбить текст s на $|A|$ частей (назовём это разбиение S) так, чтобы отличие между A и S было минимальным. Отличием двух наборов одинаковой длины A и S назовём значение $\sum_{i=1}^{|A|} dist(A_i, S_i)$, где $dist(A_i, S_i)$ — расстояние Левенштейна между строками A_i и S_i .

Важно, что разрезать текст вы можете только по пробелам, причём этот пробел не будет входить ни в один из отрезков (то есть он пропадает). То есть разбиение S можно описать количеством слов (подстрок между двумя пробелами или концами исходной строки) в каждой строке разбиения.

Для лучшего понимания, рассмотрим пример: Пусть текст s равен "when the imposter is sus". Также пусть преобразование через аудио дало нам набор A из двух строк: первая равняется "when composter", а вторая равняется "has bus". Тогда, если мы разделим текст s так, чтобы в первой строке было 3 слова, а во второй — 2, то набор S будет равняться ["when the imposter", "is sus"]. Тогда отличием наборов A и S будет значение $dist("when the imposter", "when composter") + dist("is sus", "has bus")$. Посчитав расстояния Левенштейна, получим, что отличие равно $5 + 3 = 8$.

В первой строке входных данных записано число t — количество наборов входных данных. Затем идут описания наборов.

В первой строке набора даётся одно число n — количество слов в тексте s .

Во второй строке набора даётся текст s , содержащий n слов.

В третьей строке содержится одно число $|A|$ — количество строк в разбиении.

Далее каждая строка из разбиения A описывается двумя строками ввода: в первой содержится количество слов в этой строке, а во второй — сама строка.

Для каждого набора входных данных выведите $|A|$ **положительных** чисел — количество слов в данной строке вашего разбиения S . Также сумма этих чисел должна равняться количеству слов в строке s .

Оценка за каждый набор входных данных будет вычисляться по формуле $5 \cdot \frac{\text{jury_diff}}{\text{participant_diff}}$, где participant_diff — отличие A и разбиения участника, а jury_diff — минимальное отличие, которое смогло получить жюри или участники. Оценкой за тест будет сумма оценок за все наборы входных данных.

В первом тесте $t = 6$. Длина текста s в каждом наборе входных данных не превышает 1000. Оценка за этот тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 14$. Длина текста s в каждом наборе входных данных не превышает 10^4 . Во время тура проверяется, что сумма выведенных чисел совпадает с количеством слов в строке. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

Входные данные	Результат
1 5 when the imposter is sus 2 2 when composter 2 has bus	3 2

Задача E. Bubble sort

Перед вами поставили n стаканов, изначально в каждом из них может быть либо m , либо 0 разноцветных шариков.

Стаканы достаточно узки, поэтому шарики находятся один над другим — и в любой момент времени достать можно только самый верхний. Иначе говоря, состояние стакана в любой момент времени представимо в виде массива чисел: a_1, \dots, a_t , где a_i — это цвет i -го шарика в стакане (нумерация начинается снизу).

Ваша задача — сделать так, чтобы шарики одного цвета оказались в одном стакане.

Пусть вы хотите переложить верхний шар из стакана i в стакан j . Вы можете это сделать только в случае, если выполнено одно из двух условий:

- Стакан j пустой
- В стакане j сейчас $\leq m-1$ шар, и цвет верхнего шара j -го стакана совпадает с цветом верхнего шара i -го стакана

В первой строке входного файла указано число t - количество наборов входных данных в файле. Описание каждого набора начинается со строки, содержащей числа n и m — количество стаканов и максимально возможное количество шариков в одном стакане соответственно.

В последующих n строках указано содержание стаканов в формате: Сначала указано число c_i - количество элементов в текущем стакане, после чего указаны c_i чисел - цвета шариков в стакане, в порядке от самого нижнего до самого верхнего. Выведите t решений для наборов входных данных в следующем формате:

В первой строке каждого решения набора данных выведите число k — количество действий для сортировки в Вашем решении. В следующих k строках выведите сами действия: по два числа (x_i, y_i) — операция перекладывания верхнего шара из стакана x_i в стакан y_i .

Оценка решения вычисляется по следующей формуле:

Введем функцию $f(solution) = \sum_{i=1}^n cnt_i^2$. Здесь cnt_i - количество различных элементов в итоговом стакане.

Введем функцию $g(solution) = \frac{m\sqrt{n-\#empty} - \sqrt{f(solution) - n + \#empty}}{m\sqrt{n-\#empty}}$, где $\#empty$ - количество пустых стаканов.

Оценкой за решение одного набора входных данных будет величина $10 \cdot \left(\frac{g(solution)}{g(jury_solution)}\right)^3$, $jury_solution$ - это лучшее решение среди всех участников и решения жюри.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. Баллы начисляются только в случае, если все выведенные ходы во всех тестах можно сделать. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. Баллы начисляются только в случае, если все выведенные ходы во всех тестах можно сделать. Во время тура проверяется только возможность сделать ходы, описанные во входном файле. Проверка правильности ответа осуществляется в режиме offline (результат виден после окончания тура).

Примеры

1	5
4 3	2 4
3 1 2 3	3 4
3 2 2 1	1 3
3 3 3 1	1 2
0	1 4

Задача F. Глеб и го

Глеб обожает играть в го. Партия в го проходит на квадратной клетчатой доске размера $n \times n$, на которую выкладываются белые и чёрные камни (в этой задаче, для удобства, камни будут находиться внутри клеток, хотя в оригинальной игре они выкладываются на пересечения линий доски). Проблема в том, что партия в го играется между двумя игроками, а у Глеба, к сожалению, совсем нет друзей (которые умеют играть в го). Так как поиграть Глебу все-таки хотелось, он придумал свои правила для го, в которых играет один человек, и предложил вам сыграть по его правилам.

Дана доска размера $n \times n$, каждая клетка которой либо пустая, либо содержит чёрный камень. В вашем распоряжении есть k белых камней, которые вы можете выложить на доску.

Говоря про соседние клетки в дальнейшем будем иметь в виду соседей сверху, снизу, справа или слева. Все камни одного цвета на доске делятся на **группы**: два одноцветных камня находятся в одной группе, если от одного из них можно прийти до другого, перемещаясь по соседним клеткам, содержащим камни того же цвета (в частности, один камень, не имеющий соседних клеток с камнями того же цвета, образует группу из одного камня).

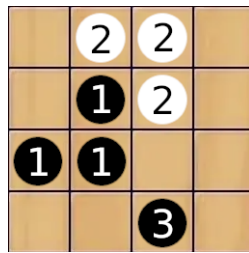


Рис. 1: На доске пронумерованы 3 группы камней: две чёрные и одна белая

Группа камней считается **живой**, если хотя бы у одного камня из группы есть соседняя пустая клетка; в противном случае группа камней считается **мёртвой**.

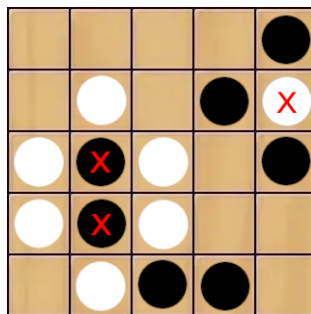


Рис. 2: Мёртвые камни на доске выделены крестиком

Вы можете выложить на доску **не более** k белых камней, класть камень можно только в пустую клетку, в одной клетке не может находиться два камня. После того, как вы их выложите, происходит подсчёт очков: сначала с доски снимаются все мёртвые белые группы камней, после чего снимаются оставшиеся мёртвые чёрные группы камней (заметьте, что некоторые чёрные группы могут «ожить» после снятия белых камней, и сняты уже не будут). Результатом игры для вас является количество снятых с доски чёрных камней, и ваша задача — максимизировать его.

В первой строке входных данных записано число t — количество наборов входных данных. Далее следуют описания наборов.

В первой строке набора вводятся два числа n и k — размер доски и количество доступных белых камней. Далее вводятся n строк, по n символов в каждой, описывающие состояние доски: клетка с координатами (i, j) либо пустая (в строке i на позиции j стоит символ "."), либо содержит чёрный камень (в строке i на позиции j стоит символ "*").

Для каждого набора в первой строке выведите число $m \leq k$ — количество использованных белых камней. В последующих m строках выведите через пробел координаты клеток, в которые вы поставили белые камни, в произвольном порядке. Координаты клеток нумеруются с единицы, все выведенные координаты должны быть различными, на месте выведенных координат не должно быть чёрных камней.

В первом тесте $t = 3$. Оценка за этот тест: 30 баллов. За каждый правильно решённый набор вы получите 10 баллов. Проверка осуществляется в режиме online (результат виден сразу).

Во втором тесте $t = 7$. Оценка за этот тест: 70 баллов. За каждый правильный набор можно получить до 10 баллов. Оценка за каждый набор будет вычисляться по формуле $10 \cdot \left(\frac{\text{participant_ans}}{\text{jury_ans}}\right)^7$, где jury_ans — максимальное количество снятых чёрных камней в решениях жюри и всех участников, а participant_ans — количество снятых чёрных камней в решении участника. Проверка осуществляется в режиме offline (результат виден после окончания тура).

Примеры

Входные данные	Результат
1	5
4 5	1 2
* . . *	2 2
* . . .	3 1
. . * *	1 3
. . . .	2 4