

Общая информация по задачам второго тура

Доступ к результатам проверки решений задач во время тура

Во всех задачах вы можете неограниченное число раз запрашивать результат окончательной проверки. Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.

Ограничение на размер исходного кода программы-решения

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Процесс тестирования

Во всех задачах баллы за подзадачу начисляются только при прохождении всех тестов подзадачи. В таблице системы оценивания для каждой подзадачи указаны номера *необходимых подзадач*. Тесты для данной подзадачи запускаются только, если все необходимые подзадачи пройдены.

Ограничения

Задача	Ограничение по времени	Ограничение по памяти
5. Управление видеонаблюдением	1 секунда	512 МБ
6. Расшифровка ДНК	2 секунды	512 МБ
7. Курьерская служба	2 секунды	512 МБ
8. Тренажёр «10 ₂ -пальцевый набор»	3 секунды	512 МБ

Замечания

В задачах с большим размером входных данных жюри рекомендует участникам, использующим языки группы C/C++, отправлять решения на проверку с использованием компилятора LINUX GNU C++ и считывать данные, используя `scanf`.

Обратите внимание, что на этом туре доступна возможность отправки решений на Java и Python, а также решений интерактивной задачи под операционной системой Ubuntu Linux.

Задача 5. Управление видеонаблюдением

Имя входного файла: video.in
Имя выходного файла: video.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Охранное агентство получило заказ на обеспечение видеонаблюдения двух зданий. В каждом из зданий установлено множество видеокамер.

На одной из стен зала видеонаблюдения расположена панель, представляющая собой прямоугольник, состоящий из n горизонтальных рядов по m мониторов в каждом. На каждый монитор выводится изображение с камеры, находящейся в одном из зданий. Зал видеонаблюдения оборудован инновационным пультом управления с четырьмя кнопками: «влево», «вправо», «вверх» и «вниз».

Кнопка «влево» перемещает изображение с каждого монитора на монитор, находящийся слева от него. При этом изображение из самого левого монитора в каждом ряду перемещается на самый правый монитор этого ряда.

Аналогичным образом действуют кнопки «вправо», «вверх» и «вниз». Кнопка «вправо» перемещает изображение с каждого монитора на монитор, находящийся справа от него. Изображения из самого правого монитора в каждом ряду перемещаются на самый левый монитор этого ряда. Кнопка «вверх» перемещает изображение с каждого из мониторов на монитор, находящийся над ним. Изображения из самого верхнего ряда перемещаются на мониторы самого нижнего ряда. Кнопка «вниз» перемещает изображение с каждого из мониторов на монитор, находящийся под ним. Изображения из самого нижнего ряда перемещаются на мониторы самого верхнего ряда.

Назовём блок мониторов размером 2×2 *удобным для наблюдения*, если эти мониторы показывают изображения из одного и того же здания. В результате перемещения изображений по командам с пульта управления количество удобных для наблюдения блоков может изменяться. При этом один и тот же монитор может входить в несколько удобных для наблюдения блоков.

Требуется написать программу, определяющую максимальное количество удобных для наблюдения блоков, которое можно получить, управляя мониторами с пульта.

Формат входных данных

Первая строка входных данных содержит два целых числа: n — количество рядов и m — количество мониторов в каждом ряду ($2 \leq n, m \leq 1000$). Следующие n строк описывают ряды мониторов в порядке сверху вниз. Каждая из этих строк содержит по m символов, описывающих мониторы в соответствующем ряду в порядке слева направо. Символ «1» означает, что на монитор выводится изображение из первого здания, а символ «2» — из второго здания.

Формат выходных данных

Выходные данные должны содержать единственное целое число — максимальное количество удобных для наблюдения блоков, которые можно получить, перемещая изображения на мониторах.

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения	Необх. подзадачи
		n, m	
1	37	$2 \leq n, m \leq 50$	
2	28	$2 \leq n, m \leq 300$	1
3	35	$2 \leq n, m \leq 1000$	1, 2

Примеры

video.in	video.out
2 4 1221 1221	2
3 2 22 22 22	2
3 3 111 121 111	3

Пояснения к примерам

В первом примере с помощью команды «вправо», можно получить слева удобный для наблюдения блок из единиц, а справа — удобный для наблюдения блок из двоек.

Во втором примере изначально на мониторе присутствуют два удобных для наблюдения блока.

В третьем примере, например, командами «вправо» и «вниз» можно добиться наличия трёх удобных для наблюдения блоков из единиц.

Задача 6. Расшифровка ДНК

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

При проведении раскопок на территории Республики Татарстан были обнаружены останки неизвестного древнего животного, обитавшего в окрестностях современной Казани миллионы лет назад. Как и у всех живых организмов, молекула его ДНК представляет собой последовательность из n нуклеотидов, однако число встречающихся в ней различных нуклеотидов может отличаться от современных организмов.

Для изучения находки был создан специальный прибор, который может просканировать последовательный участок нуклеотидов в ДНК и вычислить, сколько различных видов нуклеотидов содержится на нём. К сожалению, молекула ДНК может выдержать не более q операций сканирования, после чего разрушается.

Исследователи хотят с помощью этого прибора найти k — количество различных нуклеотидов в ДНК, и определить позиции, в которых в ДНК находятся одинаковые нуклеотиды. Ученые хотят закодировать последовательность нуклеотидов в молекуле последовательностью целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), таких что одинаковые числа кодируют одинаковые нуклеотиды, а различные числа — различные нуклеотиды.

Требуется написать программу, которая, взаимодействуя с программой жюри, определит количество различных нуклеотидов в последовательности, а также числовую последовательность, кодирующую последовательность нуклеотидов в молекуле ДНК.

Формат взаимодействия с тестирующей системой

При запуске решения на вход подается целое число n — длина молекулы ДНК ($1 \leq n \leq 3000$).

Для каждого теста зафиксированы числа k — количество различных нуклеотидов ($1 \leq k \leq n$) и q — максимальное количество запросов. Гарантируется, что q запросов достаточно, чтобы решить задачу. Эти числа не сообщаются программе участника, но ограничения на эти числа в различных подзадачах приведены в таблице системы оценивания. Если программа участника делает более q запросов программе жюри, на этом тесте она получает в качестве результата тестирования «Неверный ответ».

Чтобы сделать запрос, следует вывести строку «? i j », где i и j — целые положительные числа, номера первого и последнего нуклеотида непрерывного участка молекулы ДНК, для которого требуется узнать число различных нуклеотидов в нём ($1 \leq i \leq j \leq n$).

В ответ на каждый запрос программа получает целое число p — количество различных нуклеотидов в фрагменте ДНК, указанном в запросе.

Если программа определила ответ на задачу, то она должна вывести три строки. Первая строка должна содержать слово «Ready!». Вторая строка должна содержать целое число k — количество различных нуклеотидов в молекуле. Третья строка должна содержать последовательность n целых чисел, разделенных пробелами: a_1, a_2, \dots, a_n — коды нуклеотидов ($1 \leq a_i \leq k$). Если подходящих последовательностей несколько, то допускается вывести любую из них.

После этого программа должна завершиться.

Примеры

стандартный ввод	стандартный вывод
2 2	? 1 2 Ready! 2 1 2
3 1 2	? 1 2 ? 1 3 Ready! 2 1 1 2

Пояснения к примерам

В первом примере $n = 2$, за один запрос можно определить, равны ли первый и второй нуклеотид друг другу.

В втором примере $n = 3$, результат первого запроса показывает, что первые два нуклеотида одинаковые, результат второго запроса позволяет сделать вывод о том, что третий нуклеотид от них отличается. В этом случае допустимы два варианта ответа: 1 1 2 и 2 2 1.

В точности соблюдайте формат выходных данных. После каждого вывода обязательно выводите один перевод строки и сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения			Необх. подзадачи
		n	k	q	
1	20	$1 \leq n \leq 300$	$1 \leq k \leq 2$	$q = 72\,000$	
2	25	$1 \leq n \leq 300$	$1 \leq k \leq n$	$q = 72\,000$	1
3	25	$1 \leq n \leq 3\,000$	$1 \leq k \leq 10$	$q = 72\,000$	1
4	15	$1 \leq n \leq 3\,000$	$1 \leq k \leq n$	$q = 72\,000$	1-3
5	15	$1 \leq n \leq 3\,000$	$1 \leq k \leq n$	$q = 36\,000$	1-4

Задача 7. Курьерская служба

Имя входного файла:	twopaths.in
Имя выходного файла:	twopaths.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В корпорации работают n сотрудников, один из которых является директором. У каждого сотрудника, кроме директора, имеется ровно один непосредственный руководитель.

Каждый сотрудник выполняет свою строго определённую работу. Если сотруднику a требуется выполнить работу, которой занимается сотрудник b , то он должен отправить сотруднику b заявку. Согласно корпоративной политике напрямую заявки могут передаваться только либо от сотрудника его непосредственному руководителю, либо наоборот — от руководителя к непосредственному подчинённому. Заявка передается от сотрудника к сотруднику до тех пор, пока не достигнет сотрудника b .

Чтобы разгрузить сотрудников, корпорация наняла k курьеров. Курьер с номером i закреплён за парой сотрудников a_i и b_i и занимается доставкой заявок между ними. Если одному сотруднику из этой пары требуется передать заявку другому, то он отдаёт заявку курьеру, который последовательно передаёт заявку между сотрудниками в соответствии с корпоративной политикой, пока заявка не пройдет всех необходимых промежуточных сотрудников и окажется доставленной адресату. В процессе доставки одной заявки курьер не посещает одного и того же сотрудника более одного раза.

С целью оптимизации расходов решено найти пару курьеров, пути следования которых перекрываются наиболее сильно, и отказаться от услуг одного из них, поручив его работу второму. Назовём *степенью дублирования* для пары курьеров количество переходов между сотрудником и его непосредственным начальником в любом направлении, входящих в путь как первого, так и второго курьера.

Требуется написать программу, которая определит двух курьеров с наибольшей степенью дублирования.

Формат входных данных

Первая строка входных данных содержит два целых числа: n — количество сотрудников и k — количество курьеров ($2 \leq n, k \leq 2 \cdot 10^5$). Сотрудники пронумерованы от 1 до n , директор имеет номер 1.

Вторая строка содержит $(n - 1)$ целых чисел: p_2, p_3, \dots, p_n — номера непосредственных руководителей каждого сотрудника, кроме директора ($1 \leq p_i < i$).

Следующие k строк содержат по два целых числа: a_i и b_i — номера сотрудников в паре, за которой закреплён курьер с номером i ($1 \leq a_i, b_i \leq n; a_i \neq b_i$). За одной и той же парой сотрудников может быть закреплено несколько курьеров.

Формат выходных данных

Первая строка выходных данных должна содержать единственное целое число — наибольшую степень дублирования двух курьеров.

Вторая строка выходных данных должна содержать два различных целых числа от 1 до k — номера курьеров, входящих в пару с наибольшей степенью дублирования. Если таких пар несколько, можно вывести любую из них.

Замечание

Будем говорить, что сотрудник a *подчиняется приказам* сотрудника b , если b является непосредственным начальником a , или же непосредственный начальник a подчиняется приказам сотрудника b . Естественно называть *важным* курьера, передающего заявки от сотрудника к кому-либо, кто подчиняется его приказам (или в обратном направлении).

Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения			Необх. подзадачи
		n	k	Дополнительно	
1	29	$2 \leq n \leq 100$	$2 \leq k \leq 100$	—	
2	12	$2 \leq n \leq 4000$	$2 \leq k \leq 1000$	—	1
3	7	$2 \leq n \leq 10^5$	$2 \leq k \leq 1000$	—	1-2
4	8	$2 \leq n \leq 10^5$	$2 \leq k \leq 5000$	—	1-3
5	10	$2 \leq n \leq 10^5$	$2 \leq k \leq 50\,000$	Число переходов между любым сотрудником и директором не превышает 20	
6	12	$2 \leq n \leq 10^5$	$2 \leq k \leq 50\,000$	Каждый курьер является важным	
7	12	$2 \leq n \leq 10^5$	$2 \leq k \leq 50\,000$	—	1-6
8	10	$2 \leq n \leq 2 \cdot 10^5$	$2 \leq k \leq 2 \cdot 10^5$	—	1-7

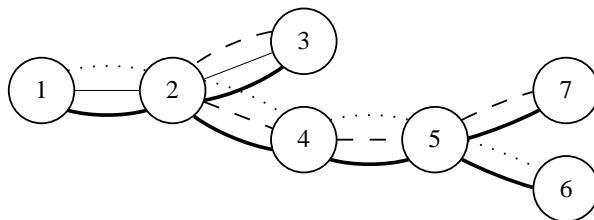
Примеры

twopaths.in	twopaths.out
4 2 1 2 2 1 3 1 4	1 2 1
4 2 1 2 3 1 2 3 4	0 1 2
7 3 1 2 2 4 5 5 1 3 3 7 6 1	2 2 3
4 3 1 2 3 1 4 4 1 1 4	3 2 1

Пояснения к примерам

В первом примере имеется два курьера. Первый курьер доставляет заявки от сотрудника 1 сотруднику 3 или обратно. Так, доставляя заявку от сотрудника 1 к сотруднику 3, он сначала передает заявку от сотрудника 1 сотруднику 2, а затем от сотрудника 2 сотруднику 3. Второй курьер доставляет заявки от сотрудника 1 сотруднику 4 или обратно. Например, доставляя заявку от сотрудника 4 сотруднику 1, он сначала передаёт её от сотрудника 4 сотруднику 2, а затем от сотрудника 2 сотруднику 1. Степень дублирования этой пары курьеров равна 1, так как для них общим является переход между сотрудником 1 (директором) и сотрудником 2.

Во втором примере также имеется два курьера, но их пути не пересекаются.



Третий пример изображён на рисунке выше. Жирными линиями обозначены переходы между сотрудниками. Тонкими линиями обозначен путь первого курьера между сотрудниками 1 и 3. Длинным пунктиром обозначен путь второго курьера между сотрудниками 3 и 7. Коротким пунктиром обозначен путь третьего курьера между сотрудниками 1 и 6.

В пути первого и второго курьера одновременно входит только переход между сотрудниками 2 и 3. В пути первого и третьего курьера одновременно входит только переход между сотрудниками 1 и 2. В пути второго и третьего курьера одновременно входят два перехода: между сотрудниками 2 и 4 и между сотрудниками 4 и 5. Таким образом, пара из курьеров с номерами 2 и 3 имеет наибольшую степень дублирования, равную 2.

В четвертом примере все курьеры переносят заявки между директором и сотрудником номер 4. Таким образом, все пары курьеров имеют степень дублирования равную 3. Разрешается вывести любую пару.

Задача 8. Тренажёр «10₂-пальцевый набор»

Имя входного файла: `typing.in`
Имя выходного файла: `typing.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Современный робот-программист должен владеть слепым 10₂-пальцевым методом набора бинарных строк, состоящих из символов «0» и «1». В инновационном тренажёре, созданном специально для уже освоивших 10₂-пальцевый набор роботов, предлагается следующее упражнение. В верхней части экрана выводится строка, состоящая из нулей и единиц. Ниже выводятся пары (c_i, w_i) , каждая из которых состоит из бинарного слова w_i и его *стоимости* c_i — количества штрафных баллов, начисляемых за каждое использование слова w_i при наборе строки.

Робот должен набрать заданную строку в виде последовательности записанных подряд префиксов или суффиксов предложенных ему бинарных слов. Одно и то же слово можно использовать произвольное количество раз, но за каждое использование префикса или суффикса начисляются штрафные баллы, равные стоимости этого слова.

Префиксом слова называется последовательность подряд идущих символов этого слова, начинающаяся с первого символа слова, а *суффиксом* — последовательность подряд идущих символов этого слова, заканчивающаяся последним символом слова. Слово целиком является как своим префиксом, так и своим суффиксом.

Требуется написать программу, которая вычисляет минимально возможное суммарное количество штрафных баллов, начисляемых роботу за набор заданной строки с использованием префиксов и суффиксов предложенных бинарных слов, или определяет, что строку набрать невозможно.

Формат входных данных

Первая строка входных данных содержит три целых числа m , n и L — длину заданной строки, количество слов, префиксы и суффиксы которых можно использовать при её наборе, и суммарную длину этих слов ($1 \leq m \leq 300\,000$; $1 \leq n \leq 300\,000$; $1 \leq L \leq 300\,000$).

Во второй строке находится заданная строка, состоящая из m символов «0» или «1». Следующие n строк описывают предлагаемые для использования бинарные слова. Сначала указывается целая стоимость слова в штрафных баллах c_i ($1 \leq c_i \leq 10^9$). Затем в той же строке через пробел следует непустое слово, состоящее из символов «0» или «1». Длина каждого слова не превосходит значения l_{max} , дополнительные ограничения на которое накладываются в некоторых подзадачах.

Формат выходных данных

Выходные данные должны содержать одно целое число — минимальное количество штрафных баллов, которое потребуется, чтобы набрать заданную строку, или число -1 , если требуемым образом набрать её невозможно.

Примеры

<code>typing.in</code>	<code>typing.out</code>
9 2 8 000110100 1 100 1 11001	4
9 3 10 010110101 3 0101 10 011 2 100	8
3 1 3 100 1 101	-1

Пояснения к примерам

В первом примере можно сначала набрать суффикс первого слова длины два, затем его же суффикс длины один, далее префикс второго слова длины три после чего первое слово целиком.

Таблица системы оценивания

В таблице системы оценивания этой задачи указаны только дополнительные ограничения, накладываемые на различные параметры входных данных. Значение l_{max} задает максимальную длину каждого из предложенных роботу бинарных слов, префиксы и суффиксы которых он может использовать.

Номер подзадачи	Баллы	Ограничения					Необх. подзадачи
		m	n	L	c_i	l_{max}	
1	20	$m \leq 50$	$n \leq 50$	$L \leq 500$	$c_i \leq 1000$	$l_{max} \leq 50$	
2	10	$m \leq 5000$	—	$L \leq 5000$	—	$l_{max} \leq 1000$	1
3	8	$m \leq 10\,000$	—	$L \leq 50\,000$	—	$l_{max} \leq 1000$	1, 2
4	8	$m \leq 50\,000$	—	$L \leq 50\,000$	—	$l_{max} \leq 2000$	1–3
5	10	$m \leq 50\,000$	$n \leq 20$	$L \leq 50\,000$	—	—	
6	5	$m \leq 50\,000$	$n \leq 200$	$L \leq 50\,000$	—	—	5
7	9	$m \leq 50\,000$	—	$L \leq 50\,000$	$c_i = 1$	—	
8	5	$m \leq 50\,000$	—	$L \leq 50\,000$	$c_i \leq 10$	—	7
9	5	$m \leq 50\,000$	—	$L \leq 50\,000$	$c_i \leq 100$	—	7, 8
10	5	$m \leq 50\,000$	—	$L \leq 50\,000$	—	—	1–9
11	5	$m \leq 100\,000$	—	$L \leq 100\,000$	—	—	1–10
12	5	$m \leq 200\,000$	—	$L \leq 200\,000$	—	—	1–11
13	5	$m \leq 300\,000$	—	$L \leq 300\,000$	—	—	1–12