

Задача А. Отметки на подмножествах

Имя входного файла: *стандартный ввод*

Имя выходного файла: *стандартный вывод*

Ограничение по времени: 2 секунды

Ограничение по памяти: 512 мегабайт

Рассмотрим множество S , состоящее из n элементов — натуральных чисел $1, 2, \dots, n$.

Сперва отметим несколько подмножеств S , а также все подмножества этих подмножеств.

Затем снимем все отметки, если они есть, с нескольких подмножеств S , а также со всех их подмножеств.

Найдите количество отмеченных подмножеств после всех этих операций.

Формат входных данных

В первой строке входных данных заданы через пробел три целых числа n , x и y . Следующие x строк содержат описания подмножеств, отмеченных на первом шаге, по одному на строке; также были отмечены все подмножества этих подмножеств. Наконец, последние y строк входных данных содержат описания подмножеств, с которых сняли отметки на втором шаге, по одному на строке; также были сняты отметки со всех их подмножеств. Описание каждого подмножества имеет вид $k \ a_1 \ a_2 \ \dots \ a_k$, где k — количество элементов данного подмножества ($0 \leq k \leq n$), а a_i — сами элементы (a_i попарно различны, $1 \leq a_i \leq n$). Элементы могут быть перечислены в любом порядке.

Формат выходных данных

В первой строке выведите одно число — количество отмеченных подмножеств после всех описанных операций.

Ограничения

- $1 \leq n \leq 10$
- $0 \leq x, y \leq 1000$

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1 1 1 1 0	1
2 0 1 2 2 1	0
3 2 1 2 1 2 2 2 3 2 1 3	3

Пояснения к примерам

В первом примере на первом шаге ставится отметка на подмножество $\{1\}$ и на пустое подмножество, на втором шаге с пустого подмножества снимается отметка.

Во втором примере отметок нет.

В третьем примере на первом шаге отмеченными оказываются следующие шесть подмножеств: $\{\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$ и $\{2, 3\}$. На втором шаге снимаются отметки с трёх подмножеств $\{\}$, $\{1\}$ и $\{3\}$.

Задача В. Отметки на подмножествах 2

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

Условие такое же, как у задачи А.

Ограничения

- $1 \leq n \leq 20$
- $0 \leq x, y \leq 1000$

Задача С. Отметки на подмножествах 3

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

Условие такое же, как у задачи А.

Ограничения

- $1 \leq n \leq 26$
- $0 \leq x, y \leq 1000$

Задача D. Гиперкуб

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

Гиперкуб — это обобщение понятия трёхмерного куба на N измерений. Нуль-мерным гиперкубом является точка, одномерным — отрезок, двумерным — квадрат. В общем же случае N -мерный гиперкуб — это правильный N -мерный многогранник, каждая из $2 \cdot N$ граней которого является $(N - 1)$ -мерным гиперкубом. Например, для $N = 2$ квадрат — это правильный многоугольник, каждая из $2 \cdot 2 = 4$ сторон которого — отрезок, то есть одномерный гиперкуб. Отметим, что N -мерный гиперкуб имеет 2^N вершин.

Старшеклассник Петя долго разбирался, что же такое гиперкуб, но наконец понял, как этот объект устроен, и ему настолько понравилось, что он даже придумал свою собственную игру на гиперкубе. Игра заключается в следующем.

Рассмотрим N -мерный единичный гиперкуб. Расположим его таким образом, чтобы одна из вершин находилась в начале координат — точке $(0, 0, \dots, 0)$ в N -мерном пространстве, а для любой из остальных вершин каждая координата равнялась бы нулю или единице. В каждой из 2^N вершин запишем по целому неотрицательному числу. Игрок начинает свой путь в начале координат. За один ход можно переместиться из текущей вершины по любому ребру при условии, что сумма координат новой вершины строго больше суммы координат старой. Игра заканчивается, когда игрок попадает в вершину $(1, 1, \dots, 1)$, имеющую максимальную сумму координат — N . Результат игры — сумма чисел во всех посещённых игроком вершинах. Цель игры — пройти по гиперкубу таким образом, чтобы эта сумма (количество очков за игру) оказалась как можно больше.

Петя довольно быстро понял, что между двумя вершинами гиперкуба A и B ребро есть тогда и только тогда, когда все координаты этих вершин (A_1, A_2, \dots, A_N) и (B_1, B_2, \dots, B_N) совпадают, кроме одной, которая равна нулю у одной из вершин (скажем, A) и единице у другой (B). Поскольку при этом $A_1 + A_2 + \dots + A_N + 1 = B_1 + B_2 + \dots + B_N$, то по такому ребру можно перемещаться из A в B , но не наоборот. Однако, сыграв в свою игру, Петя не может с уверенностью сказать, является ли полученная им сумма максимальной, или можно на данном гиперкубе сыграть по-другому и набрать больше очков.

Напишите программу, которая по данному гиперкубу находит максимальную сумму, которую можно получить, сыграв в эту игру.

Формат входных данных

В первой строке записано число N ($1 \leq N \leq 10$) — размерность гиперкуба. В следующих 2^N строках содержится по одному числу в каждой; в $(k + 2)$ -й строке записано C_k ($0 \leq C_k \leq 1000$) — число в вершине с номером k .

Номер вершины вычисляется так: вершина A с координатами (A_1, A_2, \dots, A_N) имеет номер, равный $A_1 \cdot 2^{N-1} + A_2 \cdot 2^{N-2} + \dots + A_{N-1} \cdot 2 + A_N$, то есть координаты просто интерпретируются как двоичная запись номера вершины. В этой нумерации начальная вершина имеет номер 0, а конечная — номер $2^N - 1$.

Формат выходных данных

Выведите одно число — максимальную сумму, которую можно получить при игре на данном гиперкубе.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3 4 5 6 7 8	21

Пояснение к примеру

Наш маршрут таков:

- вершина 0 (число 1, координаты $(0, 0, 0)$) — начальная
- вершина 4 (число 5, координаты $(1, 0, 0)$)
- вершина 6 (число 7, координаты $(1, 1, 0)$)
- вершина 7 (число 8, координаты $(1, 1, 1)$) — конечная

Наше количество очков: $1 + 5 + 7 + 8 = 21$.

Любой другой маршрут с соблюдением правил игры даёт меньшее количество очков.

Задача Е. Плитки

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

У Пети имеется неограниченный набор красных, синих и зелёных плиток размером 1×1 . Он выбирает ровно N плиток и выкладывает их в полоску. Например, при $N = 10$ она может выглядеть следующим образом:

К	К	К	С	З	К	К	З	К	С
---	---	---	---	---	---	---	---	---	---

Буквой К обозначена красная плитка, С – синяя, З – зелёная. После этого Петя заполняет следующую таблицу:

	Красный	Синий	Зеленый
Красный	Y	Y	Y
Синий	Y	N	Y
Зеленый	Y	Y	N

В клетке на пересечении строки, отвечающей цвету А, и столбца, отвечающего цвету В, он записывает «Y», если в его полоске найдется место, где рядом лежат плитки цветов А и В, и «N» в противном случае. Считается, что плитки лежат рядом, если у них есть общая сторона. Очевидно, что таблица симметрична относительно главной диагонали – если плитки цветов А и В лежали рядом, то рядом лежали и плитки цветов В и А. Назовём такую таблицу *диаграммой смежности* данной полоски.

Так, данная таблица представляет собой диаграмму смежности приведённой выше полоски.

Петя хочет узнать, сколько различных полосок имеет определённую диаграмму смежности. Помогите ему.

Формат входных данных

Первая строка входных данных содержит число N ($1 \leq N \leq 100$). Следующие три строки, содержащие по три символа из набора {«Y», «N»}, соответствуют трём строкам диаграммы смежности. Других символов, включая пробелы, во входных данных не содержится. Входные данные корректны, то есть диаграмма смежности симметрична.

Формат выходных данных

Выведите количество полосок длины N , имеющих указанную диаграмму смежности.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
10 YYY YNY YYN	4596
3 YYY YYY YYY	0

Задача F. Коммивояжёр возвращается!

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

Формат входных данных

В системе Альфы Центавра n планет. Это число записано в первой строке входных данных ($1 \leq n \leq 19$). Следующие n строк содержат по n чисел каждая: j -е число на i -й из этих строк — стоимость перемещения a_{ij} от i -й планеты до j -й. Числа в каждой строке разделены пробелами. Числа a_{ii} не несут полезной информации. Все числа во входных данных положительны и не превосходят 10^8 .

Формат выходных данных

В первой строке выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите n чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	5
8 1 6	3 1 2
3 5 7	
4 9 2	

Задача G. Самый длинный путь

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

В данном ориентированном графе найдите самый длинный путь такой, что каждая вершина графа встречается в нём не более одного раза.

Формат входных данных

В первой строке заданы через пробел два целых числа n и m ($1 \leq n \leq 22$, $0 \leq m \leq 1000$). В следующих m строках заданы рёбра графа в формате $u_i v_i$ — номера начальной и конечной вершин ребра i , соответственно. Граф может содержать петли и кратные рёбра.

Формат выходных данных

В первой строке выведите длину искомого пути l . Во второй строке выведите $l + 1$ число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 2 3 3 1	2 1 2 3
4 6 1 2 2 1 2 3 2 4 3 2 4 2	2 1 2 4
5 3 3 2 2 2 1 5	1 3 2

Задача Н. Дорога

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

В Древнем государстве Оссия было два города, между которыми была проложена дорога длиной S метров. Через каждый метр стояли столбики, на каждом из которых по некоторому принципу (этот секретный принцип был известен только древним монахам Шамбалы) было написано по букве (а алфавит там у них был латинский).

Однажды князь-король Василий I решил, что человек, когда он едет по этой дороге, слишком редко вспоминает о нём. Он решил это исправить. Для этого он повелел на некоторых столбиках вместо буквы написать «Здесь был Вася». По его представлению, человек, проехав любой участок дороги длиной K метров, должен обязательно хоть раз увидеть такую надпись. Иными словами, среди каждых K идущих подряд столбиков должен оказаться хоть один, на котором буква заменена на надпись. При этом, чтобы не слишком раздражать монахов (а они люди обидчивые), Василий I приказал выбрать для надписи такие столбики, чтобы среди стёртых букв оказалось как можно меньше различных букв латинского алфавита.

Помогите боярам выполнить приказ своего повелителя.

Формат входных данных

В первой строке написано одно целое число K ($1 \leq K \leq 100\,000$). Во второй строке — без пробелов написано S заглавных латинских букв в той последовательности, в которой ими помечены столбики вдоль дороги. Гарантируется, что $K \leq S \leq 100\,000$.

Формат выходных данных

В первой строке выведите N — минимальное количество различных букв латинского алфавита, которые хотя бы на одном столбике придётся стереть, чтобы написать «Здесь был Вася». В следующих N строках выведите те заглавные буквы латинского алфавита, которые потребуется хоть раз стереть. Буквы можно выводить в любом порядке. Если ответов с минимальным N несколько, можно вывести любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 АВА	1 А
2 АВВАА	2 А В

Задача I. В поисках доспехов. . .

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

Благодаря вашим усилиям главный герой получил великолепное оружие. Но для освобождения мира этого мало. Он быстро смекнул, что ему не мешает приобрести доспехи. Естественно, его интересуют самые мощные доспехи — даэдрические. Полный комплект доспехов состоит из k элементов (например, кольчуга, шлем, щит. . .). Герой хочет собрать полный комплект даэдрических доспехов.

В мире Morrowind существует n городов. Некоторые пары городов соединены дорогами. По каждой дороге разрешено движение в обоих направлениях. Между парой городов может быть более одной дороги. Не существует дороги, ведущей из города в себя. Всего m дорог. За то, чтобы пройти по любой дороге, герой должен заплатить определённую сумму денег.

В каждом городе есть оружейный магазин. Там можно покупать составляющие комплекта. Если купить что-нибудь в некотором магазине, то на все последующие покупки в этом магазине предоставляется скидка, зависящая от магазина и покупки. Несколько скидок в одном магазине складываются: например, после покупки в каком-то магазине вещей, дающих скидки 5% и 7%, следующая покупка в этом магазине будет стоить на 12% дешевле.

Главный герой живёт в городе v . Он хочет купить все k элементов доспехов (каждого по одной штуке) и вернуться домой, потратив как можно меньше денег. Вы просто обязаны ему помочь!

Формат входных данных

В первой строке входных данных записаны целые числа n ($1 \leq n \leq 50$), m ($1 \leq m \leq 500$), k ($1 \leq k \leq 7$) и v ($1 \leq v \leq n$).

Далее следует n строк с описанием магазинов. В $(i + 1)$ -й строке находится описание магазина, расположенного в i -м городе: оно состоит из k пар целых чисел. Первое число — стоимость j -ой составляющей ($1 \leq a_{ij} \leq 10^6$, либо $a_{ij} = 0$, если купить нельзя; все a_{ij} кратны 100), а второе — скидка в процентах, если купить j -ю составляющую в i -ом магазине ($0 \leq p_{ij} \leq 10$).

Далее следует m строк с описанием дорог. Каждая строка содержит три целых числа a , b и c — номера городов, которые соединяет дорога, и стоимость проезда по ней ($1 \leq a, b \leq n$, $a \neq b$, $1 \leq c \leq 10^6$).

Формат выходных данных

Выведите единственное число — минимальную сумму, необходимую герою, либо -1 , если полный комплект доспехов собрать невозможно.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 2 1 500 0 500 0 300 10 500 5 1 2 50	850

Задача J. Половинки

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

Даша принесла своей младшей сестре Еве игрушку: детскую книжку «Половинки». В этой книжке две половины: верхняя и нижняя. Страницы в половинах можно листать независимо. Всего в книжке и сверху, и снизу по n разворотов. Развороты в каждой половине пронумерованы целыми числами от 1 до n в порядке, в котором обычно листают книгу.

Сверху на каждом развороте нарисована банка с краской. Все n цветов краски разные, и для удобства пронумерованы целыми числами от 1 до n в том же порядке, что и развороты.

Снизу на каждом развороте нарисован какой-то предмет. Каждый из n предметов имеет один из n цветов, и каждый цвет встречается ровно один раз.

Исходно книжка открыта на первом развороте сверху и первом развороте снизу. За одну секунду Ева может перевернуть одну страницу: либо сверху, либо снизу перейти либо к предыдущему развороту, либо к следующему (конечно, если он есть). Цель игры с книжкой — для каждого цвета k показать одновременно и разворот с краской цвета k сверху, и разворот с предметом цвета k снизу. Цвета можно показывать в любом порядке.

Даша задумалась: как действовать, чтобы достичь цели за минимально возможное время? Помогите ей придумать кратчайшую последовательность действий.

Формат входных данных

В первой строке задано целое число n — количество разворотов в книжке ($2 \leq n \leq 20$). Во второй строке заданы n целых чисел $a_1, a_2, a_3, \dots, a_n$ — на каком развороте снизу нарисован предмет первого, второго, третьего, ..., n -го цвета ($1 \leq a_i \leq n$, все числа a_i различны).

Формат выходных данных

Выведите строку, задающую кратчайшую последовательность действий, при которых для каждого из n цветов будет момент, когда сверху открыта банка с краской этого цвета, а снизу — предмет этого цвета. Если кратчайших ответов несколько, выведите любой из них.

Каждое действие записывается одной буквой:

- «L» — в верхней половине перевернуть страницу назад,
- «l» — в нижней половине перевернуть страницу назад,
- «R» — в верхней половине перевернуть страницу вперёд,
- «r» — в нижней половине перевернуть страницу вперёд.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 1 3 5 2 4	RrrRRlRrrLLr

Пояснение к примеру

Для примера запишем в каждую секунду, начиная с нулевой, пару: на каком по счёту развороте открыты верхняя и нижняя половинки. Жирным шрифтом выделены те пары, которые обязательно нужно получить хотя бы один раз.

$$\begin{array}{l}
 (1, 1) \\
 \xrightarrow{R} (2, 1) \xrightarrow{r} (2, 2) \xrightarrow{r} (2, 3) \\
 \xrightarrow{R} (3, 3) \xrightarrow{R} (4, 3) \xrightarrow{l} (4, 2) \\
 \xrightarrow{R} (5, 2) \xrightarrow{r} (5, 3) \xrightarrow{r} (5, 4) \\
 \xrightarrow{l} (4, 4) \xrightarrow{l} (3, 4) \xrightarrow{r} (3, 5)
 \end{array}$$