# Задача A. Сочетания

| | |
|---|---|
| Имя входного файла: | `combinations.in` |
| Имя выходного файла: | `combinations.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

*Сочетанием* из $n$ по $k$ называется возрастающий набор из $k$ чисел, каждое из которых может принимать значение от 1 до $n$. Например, есть 6 сочетаний из 4 по 2: $(1, 2)$, $(1, 3)$, $(1, 4)$, $(2, 3)$, $(2, 4)$, $(3, 4)$. Занумеруем все сочетания из $n$ по $k$, начиная с 0. Требуется найти $m$-е сочетание в этом порядке.

## Формат входного файла

Во входном файле заданы числа $n$, $k$ и $m$ ($1 \le k \le n \le 30$, $0 \le m \le 10^9$).

## Формат выходного файла

Выведите в выходной файл $m$-е в лексикографическом порядке сочетание из $n$ по $k$. Сочетания занумерованы, начиная с 0. Искомое сочетание существует.

## Пример

| combinations.in | combinations.out |
|---|---|
| 4 2 3 | 2 3 |

# Задача B. Щедрое приданое

| | |
|---|---|
| Имя входного файла: | `dowry.in` |
| Имя выходного файла: | `dowry.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Дочь короля Флатландии выходит замуж за прекрасного принца. Король хочет приготовить к их свадьбе щедрое приданое, но он не уверен, какие драгоценные камни из своей коллекции преподнести.

В коллекции короля $n$ драгоценных камней, про каждый камень известен его вес $w_i$ и его ценность $v_i$. Король хотел бы преподнести как можно более ценное приданое, но с другой стороны он хочет чтобы суммарный вес камней был *разумным*. Вес считается разумным, если он находится между $L$ и $R$ (включительно).

Помогите королю выбрать камни для приданого, чтобы их вес был разумным, а суммарная их ценность была максимальна.

## Формат входного файла

Первая строка входного файла содержит $n$ ($1 \le n \le 32$), $L$ и $R$ ($0 \le L \le R \le 10^{18}$). Следующие $n$ строк описывают драгоценные камни, каждая строка содержит два целых числа: $w_i$ и $v_i$ ($1 \le w_i, v_i \le 10^{15}$).

## Формат выходного файла

Первая строка выходного файла должна содержать число $k$ — количество камней, которое следует выбрать. Вторая строка должна содержать $k$ чисел от 1 до $n$ — номера камней (камни нумеруются, начиная с 1, в том порядке, в котором они заданы во входном файле).

Если выбрать искомое приданое невозможно, выведите $k = 0$.

## Пример

| dowry.in | dowry.out |
|---|---|
| 3 6 8<br>3 10<br>7 3<br>8 2 | 1<br>2 |

# Задача C. Уравнение

| | |
|---|---|
| Имя входного файла: | `equation.in` |
| Имя выходного файла: | `equation.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Необходимо написать программу, решающую уравнения вида

$$\frac{a \cdot x + b}{c \cdot x + d} = v.$$

Числа $a$, $b$, $c$, $d$, $v$ заданы, а $x$ — неизвестно.

## Формат входного файла

Входной файл содержит пять целых чисел: $a$, $b$, $c$, $d$, $v$. Все они не превосходят 1000 по абсолютной величине. Числа во входном файле разделены пробелами и/или переводами строк.

## Формат выходного файла

Если у указанного уравнения нет решений, выведите в выходной файл слово `NONE`. Если у уравнения ровно решение, то выведите в выходной файл строку вида `X = p/q`, где $p$ — целое число, $q$ — натуральное, $p$ и $q$ взаимно-просты, а дробь $\frac{p}{q}$ является решением уравнения. Если у уравнения более одного решения, выведите в выходной файл слово `MULTIPLE`.

## Примеры

| equation.in | equation.out |
|---|---|
| `1 2 3 4 5` | `X = -9/7` |
| `1 1 1 1 1` | `MULTIPLE` |

# Задача D. Независимое множество

| | |
|---|---|
| Имя входного файла: | `independent.in` |
| Имя выходного файла: | `independent.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Понятие последовательно-параллельного графа является очень важным в таких разделах, как электротехника, дизайн микросхем, и других. Такие графы используются для задания электрических сетей, сетей передачи данных, и т. п.

В этой задаче граф может содержать параллельные ребра.

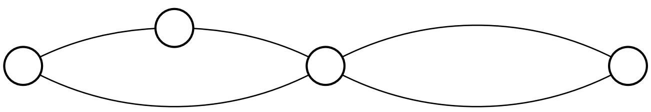
Последовательно-параллельные графы определяются рекурсивно следующим образом.

1. Граф из двух вершин $s$ и $t$, соединенных ребро $st$, является последовательно-параллельным графом с истоком $s$ и стоком $t$. Обозначим этот граф как $g$.

2. Если $G_1$ является последовательно-параллельным графом с истоком $s_1$ и стоком $t_1$, а $G_2$ является последовательно-параллельным графом с истоком $s_2$ и стоком $t_2$, то граф, получаемый объединением вершин $t_1$ и $s_2$ в одну, является последовательно-параллельным графом с истоком $s_1$ и стоком $t_2$, он обозначается $SG_1G_2$.

3. Если $G_1$ является последовательно-параллельным графом с истоком $s_1$ и стоком $t_1$, а $G_2$ является последовательно-параллельным графом с истоком $s_2$ и стоком $t_2$, то граф, получаемый объединением вершин $s_1$ и $s_2$ в одну (обозначим ее как $s_{12}$), и объединением вершин $t_1$ и $t_2$ в одну (обозначим ее как $t_{12}$) является последовательно-параллельным графом с истоком $s_{12}$ и стоком $t_{12}$, он обозначается $PG_1G_2$.

Используя приведенное выше определение, мы можем получать разнообразные последовательно-параллельные графы. На рисунке приведен в качестве примера граф $SPSgggPgg$.



Вам задан последовательно-параллельный граф. Найдите размер максимального независимого множества в нем. Множество вершин называется независимым, если оно не содержит двух вершин, соединенных ребром.

## Формат входного файла

Входной файл содержит одну строку — описание последовательно-параллельного графа. Описание представляет собой корректное выражение, содержащее только буквы ‘`S`’, ‘`P`’ и ‘`g`’. Длина выражения не превышает 100 000.

## Формат выходного файла

Выведите одно число — размер максимального независимого множества в приведенном графе.

## Пример

| independent.in | independent.out |
|---|---|
| SPSgggPgg | 2 |

# Задача E. Марсианский язык

| | |
|---|---|
| Имя входного файла: | `language.in` |
| Имя выходного файла: | `language.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Недавно ученые лаборатории радиоастрономии научно-исследовательского института формального исследования галактического антропогенеза (НИИ ФИГА) записали радиосигнал, передаваемый с Марса. После расшифровки выяснилось, что этим радиосигналом передавалась последовательность букв латинского алфавита. Этот факт не вызвал у ученых большого удивления, больше их поразило то, что на Марсе есть радиопередатчики.

Для того, чтобы прочитать переданную информацию, необходимо последовательность букв разбить на слова. Известно, что слова в марсианском языке состоят не менее, чем из $c$ и не более, чем из $C$ букв.

Назовем *качеством* слова разность между максимальным и минимальным номерами в алфавите букв, входящих в слово. Например, качество слова `ab` равно $2-1=1$, а слова `abcz` равно $26-1=25$.

Пусть была передана строка $c_1$, $c_2$, …, $c_n$. Для того, чтобы разбить ее на слова, поставим между некоторыми соседними буквами пробелы. Тогда последовательности букв, ограниченные пробелами, началом и концом строки будут являться словами. Качеством разбиения строки на слова назовем сумма качеств всех входящих в него слов.

Ваша задача — найти для строки, полученной учеными лаборатории радиоастрономии, разбиение на слова марсианского языка, обладающее максимально возможным качеством.

## Формат входного файла

Первая строка входного файла содержит три целых числа: $n$, $c$, $C$ ($1 \le c \le C \le n \le 5000$). Вторая строка входного файла содержит $n$ строчных букв латинского алфавита — переданную последовательность.

## Формат выходного файла

В первой строке выходного файла выведите качество построенного разбиения. Во второй строке выведите число $k$ — количество слов в разбиении. Далее выведите все слова разбиения — по одному в каждой из последующих $k$ строк. Конкатенация всех выведенных слов в том порядке, в котором они выведены должна быть равна исходной строке. Если возможных вариантов ответа несколько, выведите любой из них.

Если строку нельзя разбить на слова марсианского языка, то выведите в выходной файл строку `NO SOLUTION`.

## Примеры

| language.in | language.out |
|---|---|
| 7 2 4<br>abacaba | 4<br>3<br>ab<br>ac<br>aba |
| 6 4 5<br>abacab | NO SOLUTION |

# Задача F. Лексикографически минимальное число

| | |
|---|---|
| Имя входного файла: | `minlex.in` |
| Имя выходного файла: | `minlex.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Найдите число, десятичная запись которого является лексикографически минимальной среди десятичных записей всех целых чисел от $L$ до $R$.

## Формат входного файла

Входной файл содержит два целых числа: $L$ и $R$ ($1 \le L \le R \le 10^9$).

## Формат выходного файла

В выходной файл выведите ответ на задачу.

## Примеры

| minlex.in | minlex.out |
|---|---|
| 4 7 | 4 |
| 4 11 | 10 |

# Задача G. Минимум в прямоугольнике

| | |
|---|---|
| Имя входного файла: | `rmq2.in` |
| Имя выходного файла: | `rmq2.out` |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Вам задана таблица целых чисел

$$\begin{matrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{matrix}$$

и последовательность запросов $Q(x_{i1}, y_{i1}, x_{i2}, y_{i2})$. Для каждого запроса найдите минимальное значение среди значений $a_{k,l}$, где $x_{i1} \le k \le x_{i2}$ и $y_{i1} \le l \le y_{i2}$.

## Формат входного файла

Первая строка входного файла содержит числа $m$ и $n$ — размер таблицы ($1 \le m, n \le 500$). Следующие $m$ строк содержат по $n$ целых чисел каждая — $a_{ij}$. Все числа между $-2^{31}$ и $2^{31} - 1$. Далее следует $q$ — количество запросов ($1 \le q \le 200\,000$). Следующие $q$ строк содержат по четыре целых числа: $x_{i1}, y_{i1}, x_{i2}$ и $y_{i2}$ ($1 \le x_{i1} \le x_{i2} \le m$, $1 \le y_{i1} \le y_{i2} \le n$).

## Формат выходного файла

Выведите $q$ чисел — для каждого запроса выведите минимальное значение в соответствующем прямоугольнике.

## Примеры

| rmq2.in | rmq2.out |
|---|---|
| 3 3<br>3 4 5<br>2 3 4<br>5 4 3<br>3<br>1 1 2 2<br>1 2 1 3<br>1 2 3 2 | 2<br>4<br>3 |

# Задача H. Дороги

| | |
|---|---|
| Имя входного файла: | `roads.in` |
| Имя выходного файла: | `roads.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

На недавно составленном генеральном плане развития одного из районов одного из крупных городов страны $N$-мерики присутствуют $n$ асфальтовых дорог. Кроме этого, на нем присутствуют $m$ дорожных знаков.

Разумеется, перед утверждением главой администрации района план должен быть тщательно проверен. При этом одним из важных условий, которое подлежит проверке, является то, что ни один дорожный знак не находится внутри или на краю какой-нибудь дороги. Действительно, достаточно странно будет выглядеть знак «Уступи дорогу», стоящий посреди проезжей части.

Для простоты будем считать, что каждая дорога изображается на плане полосой, а каждый дорожный знак — точкой.

Пусть задана некоторая прямая $l$. *Полосой ширины $d$, симметричной относительно прямой $l$* называется геометрическая фигура, состоящая из точек, удаленных от прямой $l$ не более, чем на $d/2$.

Ваша задача состоит в том, чтобы написать программу, которая выполнит указанную выше проверку плана.

## Формат входного файла

Первая строка входного файла содержит число $n$ ($0 \le n \le 1000$). Каждая из последующих $n$ строк описывает одну дорогу. Описание каждой дороги состоит из пяти чисел: $x_1$, $y_1$, $x_2$, $y_2$, $d$. Эти числа обозначают: ($x_1$, $y_1$) и ($x_2$, $y_2$) — две различные точки прямой, относительно которой дорога симметрична, $d$ — ширина дороги ($1 \le d \le 1000$).

Далее следует строка, содержащая число $m$ ($0 \le m \le 1000$). Последующие $m$ строк содержат каждая по два числа — координаты соответствующего дорожного знака. Все координаты во входном файле целые и не превышают $10^4$ по абсолютному значению.

## Формат выходного файла

В выходной файл выведите два числа: количество дорог, внутри которых стоит хотя бы один дорожный знак, и количество дорожных знаков, которые стоят хотя бы на одной дороге.

## Примеры

| roads.in | roads.out |
|---|---|
| 1<br>0 0 0 1 2<br>3<br>0 0<br>0 10<br>10 0 | 1 2 |

# Задача I. Фабрика игрушек

| | |
|---|---|
| Имя входного файла: | `toys.in` |
| Имя выходного файла: | `toys.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Петя работает на фабрике игрушек. Фабрика производит игрушечные коляски и игрушечные автомобили.

Производство каждой игрушки состоит из двух этапов: сначала пластиковую игрушку отливают на специальном станке, а затем раскрашивают вручную. А Петя занимается тем, что переносит игрушки со станка в художественную мастерскую.

Чтобы отлить игрушку требуется $t$ секунд. Аналогично, чтобы раскрасить игрушку также требуется $t$ секунд. А вот время, которое требуется Пете, чтобы перенести игрушку от станка в художественную мастерскую, зависит от типа игрушки. Коляски легкие, и Петя переносит коляску за $c_1$ секунд. А вот автомобили — довольно тяжелые, поэтому Пете требуется целых $c_2$ секунд, чтобы перенести автомобиль. За раз Петя может переносить только одну игрушку. Без игрушки Петя бегает так быстро, что временем на его возвращение из мастерской к станку можно пренебречь.

Недавно на фабрику поступил заказ на изготовление $a_1$ игрушечных колясок и $a_2$ игрушечных автомобилей. Директор фабрики хотел бы минимизировать общее время изготовления игрушек — время от начала изготовления первой игрушки до окончания покраски последней. Помогите директору выбрать оптимальный порядок производства игрушек, при котором общее время изготовления было бы минимально.

## Формат входного файла

Входной файл содержит пять целых чисел: $t$, $c_1$, $c_2$, $a_1$, $a_2$ ($1 \le t, c_1, c_2 \le 10^4$, $c_1 \le c_2$, $0 \le a_1, a_2 \le 10^5$).

## Формат выходного файла

Выведите одно целое число — минимальное время, необходимое, чтобы произвести все необходимые игрушки.

## Пример

| toys.in | toys.out |
|---|---|
| `3 2 4 2 2` | `19` |

Можно, например, производить игрушки в таком порядке: автомобиль, коляска, коляска, автомобиль.