

## Задача А. День рождения

Имя входного файла:	birthday.in
Имя выходного файла:	birthday.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Чем старше мы становимся, тем больше начинаем ценить праздники. Ведь праздник — это возможность наконец-то увидеть своих близких, детей, внуков, друзей, хорошо провести время и получить заряд бодрости и позитивного настроения!

Наша история об одном мудром профессоре, который решил изобрести новый тип праздника. Он назвал его общим днем рождения. Вот несколько примеров, которые дал нам профессор:

- Возьмем Сережу, который родился 10 декабря 2005 года и Петю, который родился 13 декабря 2004 года, тогда 12 июня 2006 года им исполнился 1 год на двоих.
- Допустим Вова родился 1 января 1990 года, а Саша 9 января 2000 года, тогда 5 января 2005 года им было 10 лет на двоих, в 5 января 2015 будет 20 лет на двоих.
- Пусть Валера родился 1 января 2000 года, а Никита 1 июля 2000 года, тогда 2 апреля 2010 года им было 10 лет на двоих.
- Если Денис родился 31 мая 1988 года, Илья родился 6 июня 1985 года, а Вася родился 30 октября 1993 года, то 23 марта 2019 года им будет 30 лет на троих!

Подумав немного, наш дорогой профессор решил, что у него получилась отменная тема для курсовой работы. Вот как он ее описал: пусть компания из  $N$  друзей совместно отмечает дни рождения. Чтобы таких дней было больше, к общим дням рождения добавим совместные.

Для этого рассмотрим произвольное подмножество участников этой компании. Занумеруем участников подмножества номерами  $1, 2, \dots, m$  ( $m$  — произвольное, от 1 до  $n$ ). Начнем отсчет количества дней каждого из участников этого подмножества с даты рождения младшего из них. Пусть участник  $i$  ( $1 \leq i \leq m$ ) к моменту наступления дня  $k$  прожил  $d_{ik}$  дней. Тогда день  $k$  считается общим днем рождения этого набора людей, если

$$\left\lfloor \frac{\sum_{i=1}^m d_{ik}}{365 \cdot m} \right\rfloor > \left\lfloor \frac{\sum_{i=1}^m d_{ik-1}}{365 \cdot m} \right\rfloor$$

где  $\lfloor a \rfloor$  — результат округления числа  $a$  до не большего его целого. В задаче будем рассматривать только те общие дни рождения, которые являются юбилеями (т.е. кратными 10). Учитывая продолжительность жизни в нашей стране, нас интересуют только юбилеи, которые не превышают ста лет.

Вам предстоит справиться с предложенной курсовой работой, а именно — перечислить все требуемые юбилеи.

### Формат входного файла

В первой строке входного файла записано число  $N$  — количество рассматриваемых людей ( $1 \leq N \leq 14$ ). Далее в  $N$  строках идет описание каждого из них — имя и дата рождения. Имя является словом из не более чем 10 заглавных или строчных латинских букв, а дата рождения — стандартная дата в формате ДД.ММ.ГГГГ. Года нумеруются от 0001 до 3000. Внимание! Будем считать, что в феврале ВСЕГДА 28 дней, потому в тестах гарантированно не будет 29 февраля.

### Формат выходного файла

В выходной файл выведите все юбилеи для каждого подмножества людей, такие, что количество лет в среднем на рассматриваемую компанию не меньше 10 и не больше 100.

Выведите каждый общий день рождения таким образом: « <ДД.ММ.ГГГГ> <число лет> <перечисление имен людей из рассматриваемого множества> ».

Выводить общие дни рождения можно в любом порядке.

Перечислять людей из рассматриваемого множества можно в любом порядке.

## Примеры

birthday.in	birthday.out
2	05.03.1970 10 Vova.
Vova 05.03.1960	05.03.1980 20 Vova.
Denis 28.09.2004	05.03.1990 30 Vova.
	05.03.2000 40 Vova.
	05.03.2010 50 Vova.
	05.03.2020 60 Vova.
	05.03.2030 70 Vova.
	05.03.2040 80 Vova.
	05.03.2050 90 Vova.
	05.03.2060 100 Vova.
	28.09.2014 10 Denis.
	28.09.2024 20 Denis.
	28.09.2034 30 Denis.
	28.09.2044 40 Denis.
	28.09.2054 50 Denis.
	28.09.2064 60 Denis.
	28.09.2074 70 Denis.
	28.09.2084 80 Denis.
	28.09.2094 90 Denis.
	28.09.2104 100 Denis.
	17.06.2012 30 Vova Denis.
	17.06.2022 40 Vova Denis.
	17.06.2032 50 Vova Denis.
	17.06.2042 60 Vova Denis.
	17.06.2052 70 Vova Denis.
	17.06.2062 80 Vova Denis.
	17.06.2072 90 Vova Denis.
	17.06.2082 100 Vova Denis.

## Задача В. Циклы

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

Маленького Васю учат считать. Ему сказали, что он научится считать как только подсчитает результат выполнения следующей программы:

---

```
ответ = 0
перебрать  $i_1$  от 1 до  $N + a_1$ 
  перебрать  $i_2$  от 1 до  $i_1 + a_2$ 
    перебрать  $i_3$  от 1 до  $i_2 + a_3$ 
      ...
      перебрать  $i_k$  от 1 до  $i_{k-1} + a_k$ 
        ответ = ответ + 1
```

---

Конечно Вася без затруднений написал программу и нашел ответ, но она работала очень долго. А сможете ли вы посчитать ответ быстрее, чем Вася?

### Формат входного файла

Первая строка входного файла содержит 2 числа  $k$  и  $N$  — количество циклов и ограничение первого цикла соответственно ( $1 \leq k \leq 80, 1 \leq N \leq 5000$ ).  
Во второй строке содержится  $k$  чисел —  $a_1, a_2, \dots, a_k$  ( $0 \leq a_i \leq 5$ ).

### Формат выходного файла

Выведите единственно число — ответ на задачу.

### Примеры

<code>cycles.in</code>	<code>cycles.out</code>
2 3 0 1	9
3 1 1 0 0	4

## Задача С. Раскладка клавиатуры

Имя входного файла:	layout.in
Имя выходного файла:	layout.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Шел первый урок по информатике в пятом классе. Вовочка, потомственный сын программиста, сонно зевал. После того, как учительница в седьмой раз показала школьникам способ переключения раскладки на клавиатуре, глаза Вовочки не выдержали и закрылись. И провалился он в глубокий странный сон.

Видит Вовочка перед собой непонятный ноутбук. Клавиатура на нем какая-то странная — вместо стандартных пластмассовых кнопочек, светятся маленькие экранчики с разными символами, по одному на экранчик. А на мониторе неизвестная операционная система.

Повозившись полчаса с этим чудо-девайсом, он выяснил, что на клавиатуре напроочь отсутствуют клавиши «Caps Lock» и «Shift», зато есть специальная клавиша «Change Layout». Понажимав на нее несколько раз, он осознал, что эта чудо-клавиша меняет раскладку клавиатуры. Причем, каких здесь раскладок только не было: из одних цифр, из одних больших английских букв, смешанные русские и арабские буквы, раскладки, содержащие только арифметические символы и т.д.

Внезапно Вовочка осознал, что перед ним стоит ужасно важная миссия по спасению мира — нужно набрать определенный текст. Проблема в том, что символы в этом тексте почему-то из разных раскладок клавиатуры. И вот перед нашим героем стоит тяжелая проблема: «Сможет ли он спасти мир, набрав заветную фразу, переключив раскладку клавиатуры сколько угодно раз, или же мир обречен и сон закончится кошмаром?»

### Формат входного файла

Первая строка входного файла содержит число  $N$  — количество раскладок на странном ноутбуке ( $0 \leq N \leq 200$ ). Далее следуют  $N$  строк, каждая из которых содержит число  $M$  — количество символов в текущей раскладке ( $1 \leq M \leq 200$ ). А затем в этой же строке через пробел приведены сами символы текущей раскладки в количестве  $M$  штук. Каждый представленный символ является символом из таблицы ASCII с кодом от 33 до 126. Символы в раскладке могут повторяться. Заметим, что клавиши «Space», «Tab» и «Enter» были на клавиатуре чудо-девайса, соответственно данные символы точно не входят в рассматриваемые в нашей задаче.

В  $(N + 2)$ -ой строке входного файла вводится число  $L$  ( $0 \leq L \leq 1000$ ) — длина текста, который нужно набрать. В последней строке находится сам текст длиной  $L$  символов. Все символы в тексте также имеют коды ASCII от 33 до 126.

### Формат выходного файла

В выходной файл выведите одно единственное слово «YES» — если данный текст можно набрать, переключив любое количество раз раскладку на чудо-девайсе. В противном случае выведите печальное слово «NO».

## Примеры

layout.in	layout.out
3 3 a b c 6 + = ^ - * / 12 1 2 3 4 5 6 7 8 9 0 4 2 11 a^2 + b^2 = c^2	YES
3 3 a b c 6 + = ^ - * / 10 1 2 3 4 5 6 7 8 9 0 15 a^9+b^7 = z^3 -c* c	NO

## Задача D. Простая задача

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

Эта задача настолько простая, что мы даже не стали сочинять для нее историю про Васю. Вам требуется найти количество чисел  $a$  в интервале от 1 до  $n$  таких, что  $\gcd(a, k) = k$ .

Напомним, что  $\gcd(a, b)$  — наибольший общий делитель чисел  $a$  и  $b$ .

### Формат входного файла

Входной файл содержит 2 числа  $n$  и  $k$  ( $1 \leq n, k \leq 10^{18}$ ).

### Формат выходного файла

В выходной файл выведите единственное число — ответ на задачу.

### Пример

<code>numbers.in</code>	<code>numbers.out</code>
1 1	1

## Задача Е. Разбиение участников

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

На республиканскую олимпиаду по ИЗО приехали  $N$  участников. Для организации культурно-развлекательной программы требуется разделить их на две непустые группы.

Председателю оргкомитета стало интересно, а сколько существует различных способов поделить участников. Сам он не сумел справиться с поставленной проблемой и обратился к участникам олимпиады по программированию с просьбой решить задачу. Два способа разбиения считаются различными, если наборы участников хотя бы в одной из команд отличаются. Два способа, отличающиеся только перестановкой команд, считаются одинаковыми.

Напишите программу, которая по введённому числу участников определит требуемое количество способов разбить их на две группы.

### Формат входного файла

В первой строке входного файла записано целое число  $N$  — количество участников ( $1 \leq N \leq 70$ ).

### Формат выходного файла

В выходной файл выведите одно число — количество способов разбить всех участников на две непустые группы.

### Примеры

<code>partition.in</code>	<code>partition.out</code>
2	1
3	3

## Задача F. Плейлист

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

Девочка Саша очень любит слушать музыку на своём плеере. Однажды она слушала музыку так долго, что забыла сколько треков она уже прослушала. Она попросила вас помочь ей узнать ответ на этот вопрос.

Плейлист в Сашинем плере состоит из  $n$  треков, для каждого из которых известно время его исполнения. Плейлист зациклен, это значит, что после  $n$ -ой песни, будет воспроизводиться первая. Естественно, Саша слушает песни подряд, начиная с первой (функция перемешивания трека отключена).

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 1000$ ). Следующие  $n$  строк содержат по одному целому числу  $t_i$  ( $1 \leq t_i \leq 600$ ) в каждой — время исполнения  $i$ -ой песни в секундах. Последняя строка входного файла содержит число  $T$  ( $1 \leq T \leq 10^6$ ) — количество секунд, в течение которых Саша слушала музыку.

### Формат выходного файла

В первой строке выходного файла выведите целое число  $m$  — количество треков, которые прослушала Саша за  $T$  секунд. В следующей строке выведите  $m$  чисел — порядок исполнения треков на плеере. Трек, который Саша еще не дослушала до конца, тоже следует включить в ответ.

### Примеры

playlist.in	playlist.out
3 1 2 3 6	3 1 2 3
1 3 1	1 1
2 1 1 10	10 1 2 1 2 1 2 1 2 1 2



## Задача G. Пифагоровы тройки

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

Однажды на уроке математики мальчику Васе рассказали про теорему Пифагора. Вася был настолько поражен алгебраическим уравнением

$$c^2 = a^2 + b^2$$

что задался целью отыскать все его целые положительные решения!

Потратив две недели на поиск решений, Вася неожиданно понял, что их бесконечно много. Немного погрузив, он взглянул на исписанные им тетрадки и все же решил привести в порядок те решения, которые он уже нашел, а именно упорядочить их по возрастанию.

Для каждого решения он определил, что число  $a$  должно быть всегда меньше либо равно числу  $b$ . Далее он выписал все решения в столбик таким образом, что если у одной пифагоровой тройки число  $c$  меньше числа  $c$  другой пифагоровой тройки, то первая тройка выписана раньше. Если же число  $c$  первой тройки совпадает с числом  $c$  второй тройки, то он сравнивал числа  $a$  по аналогии. А если же числа  $a$  тоже совпадают, то он сравнивал числа  $b$ .

Вот какие первые 6 значений, которые он получил:

(5, 3, 4)  
(10, 6, 8)  
(13, 5, 12)  
(15, 9, 12)  
(17, 8, 15)  
(20, 12, 16)

Следуя классике жанра, поставим вопрос: сможете ли Вы сказать, какой будет  $k$ -я по счету пифагорова тройка в Васином списке?

### Формат входного файла

Входной файл содержит одно целое число  $k$  ( $1 \leq k \leq 2000$ ).

### Формат выходного файла

Выходной файл содержит три числа  $c$ ,  $a$ ,  $b$  разделенные пробелом, —  $k$ -я по счету пифагорова тройка в Васином списке.

### Примеры

<code>pythagoras.in</code>	<code>pythagoras.out</code>
6	20 12 16
20	50 30 40

## Задача Н. Проблема Васи

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

Юный плутонианец Вася прилетел на Землю для участия в программе обмена опытом. Его определили в летний детский лагерь, где он жил вместе с человеческими школьниками.

Однажды Вася получил задание проплыть по реке из пункта  $A$  в пункт  $B$ . Оказалось, что это не совсем просто. Какой-то нехороший человек, кинул в реку гравиконцентраты, из-за которых река на некоторых участках изменила скорость течения и даже, возможно, направление. Ввиду такой подлости, Вася просит вас помочь ему вычислить минимальное время, за которое он может проплыть из  $A$  в  $B$ .

### Формат входного файла

В первой строке входного файла задано четыре числа:  $n, A, B, v$  — количество гравиконцентратов, координаты начальной и конечной точек и скорость Васиного катера соответственно ( $A < B, v > 0$ ). Каждая следующая из  $n$  строчек содержит ещё по четыре числа  $l_i, r_i, vr_i, t_i$ , где  $l_i$  и  $r_i$  — координаты левого и правого концов отрезка реки ( $l_1 = 0, l_i < r_i, r_i = l_{i+1}$ ),  $vr_i$  — скорость реки на этом участке ( $0 \leq vr_i < v$ ),  $t_i$  — тип течения (0 — направление течения реки из  $B$  в  $A$ , 1 — наоборот). Точки  $A$  и  $B$  всегда принадлежат каким-то отрезкам реки.

Все числа во входном файле неотрицательные целые и не превосходят 10000. Все скорости заданы в метрах в секунду, все времена — в секундах, все расстояния — в метрах.

### Формат выходного файла

В выходной файл выведите единственное число с точностью до 6 знаков после запятой — минимальное время, за которое Вася доплывет из пункта  $A$  в пункт  $B$ .

### Примеры

river.in	river.out
1 0 10 10 0 10 0 1	1.0000000
3 55 59 42 0 47 22 0 47 59 26 1 59 70 40 0	0.0588235

## Задача I. Строка

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

Маленького Васю учат писать. Ему задали написать несколько строк текста. Он хочет затратить на это как можно меньше сил. Для этого он определил стоимость замены одной буквы на другую. Теперь у него есть строка  $S$  и строка  $T$ , в которых он хочет заменить несколько (возможно ноль) символов в строке  $S$  так, чтобы она стала подпоследовательностью строки  $T$ . При этом он хочет минимизировать суммарную стоимость замен.

Пусть дана строка  $W$  длиной  $N$ . Тогда подпоследовательностью строки называется такой набор символов:  $W_{i_1} W_{i_2} \dots W_{i_k}$ ,  $1 \leq i_j < i_{j+1} \leq N$

Вася с легкостью написал программу, теперь ваша очередь.

### Формат входного файла

Первая строка входного файла содержит непустую строку  $S$ . Вторая строка входного файла содержит строку  $T$ . Третья строка входного файла содержит число  $K$  ( $1 \leq K \leq 26$ ) — количество букв в алфавите.

Следующие  $K$  строк содержат по  $K$  чисел.  $j$ -е число на  $i$ -ой строке обозначает стоимость замены  $i$ -ой буквы алфавита на  $j$ -ю. Стоимость замены не превосходит  $10^6$  и положительна. Гарантируется, что  $i$ -е число  $i$ -ой строки равно нулю.

Длина строки  $S$  не превосходит 100 символов, строки  $T$  —  $10^4$  символов. Длина строки  $T$  всегда больше длины строки  $S$ . Строки содержат только первые  $K$  строчных букв латинского алфавита.

### Формат выходного файла

В выходной файл выведите единственное число — минимальную стоимость замен.

### Примеры

string.in	string.out
ab bacb 3 0 1 2 2 0 0 1 2 0	0
ab baa 2 0 1 3 0	3

## Задача J. Треугольники

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

Всем небезызвестный Вася увлекается различными играми. В этот раз он взял листок своей тетради в клетку и принялся рисовать различные точки в узлах клеток. Далее он решил соединить точки отрезками так, чтобы выполнялись условия:

1. Любые два отрезка могут пересекаться только в точках своих концов.
2. Нельзя добавить отрезок, не нарушив первое правило.

Вася изрисовал весь листок и заметил, что построенные отрезки образуют разбиение минимального выпуклого многоугольника (выпуклым называется многоугольник, в котором для любой стороны все остальные вершины лежат по одну сторону от прямой, проведенной через эту сторону), содержащего нарисованные Васей точки, на треугольники. Но Вася не стоит на месте, теперь его интересуют только такие разбиения, где каждый треугольник разбиения не содержит внутри своей описанной окружности других точек, нарисованных Васей.

Вася уже построил разбиение, но не умеет быстро определять, удовлетворяет ли построенное разбиение последнему условию или нет.

Помогите ему это определить.

### Формат входного файла

Входной файл в первой строке содержит два целых числа  $N$  и  $M$  — количество точек, нарисованных Васей, и количество треугольников, участвующих в разбиении ( $1 \leq N \leq 500$ ). Далее в  $N$  строках содержатся координаты нарисованных Васей точек.

Далее следует  $M$  строк по три числа в каждой  $a_i, b_i, c_i$  — номера вершин треугольника, участвующего в разбиении ( $1 \leq a_i, b_i, c_i \leq N$ ). Координаты точек во входном файле по модулю не превышают 10000.

### Формат выходного файла

Выведите “YES”, если данное разбиение удовлетворяет последнему требованию Васи, и “NO” — иначе.

## Примеры

triangle.in	triangle.out
4 2 0 0 0 1 1 0 1 1 1 2 3 2 3 4	YES
5 4 0 0 0 1 1 0 1 1 2 2 2 4 3 2 3 1 4 5 3 4 5 2	YES
4 2 0 3 3 0 3 3 1 1 1 2 4 1 2 3	NO