

## Для всех задач:

Имя входного файла: *input.txt*  
Имя выходного файла: *output.txt*  
Ограничение по памяти: *256 МБ*

### Задача 1. Вышивка кота Матроскина

Ограничение по времени на 1 тест: *1 сек.*

Как известно, кот Матроскин умеет не только готовить, но и вышивать. На этот раз он решил вышить для дяди Фёдора, уехавшего учиться в НГУ, целую картину. К сожалению, Матроскин не знает, какой ширины нужна вышивка, но он придумал алгоритм, по которому сможет всегда увеличить размер картины, не нарушая при этом рисунок.

Матроскин вышивает по канве, которая представляется прямоугольной сеткой, состоящей из линий и столбцов. В нулевом столбце не вышито ничего, в этом случае высота узора равна нулю. В первом — один крестик в нижней линии, высота узора равна 1. Дальше узор строится так. Вся готовая, вышитая часть узора длины  $2^K$  и высоты  $K$  отображается зеркально относительно вертикали и вышивается справа от этой части. Сверху над новыми столбцами в строке с номером  $K + 1$  вышивается одна линия, состоящая из крестиков. Таким образом, на этом шаге получается узор длины  $2^{K+1}$  и высоты  $K + 1$ . Вот так это выглядит для ширины 16:

								х	х	х	х	х	х	х	х
				х	х	х	х	х	х	х	х				
		х	х	х	х					х	х	х	х		
	х	х			х	х			х	х			х	х	

Ваша цель — по номеру столбца определить, какой в нём узор вышит.

#### Входные данные

В первой строке входного файла записано десятичное число  $N$  ( $0 \leq N \leq 10^9$ ) — номер столбца, для которого нужно вывести рисунок.

#### Выходные данные

В выходной файл необходимо вывести запись рисунка в заданном столбце сверху вниз (1 — крестик, 0 — пустое место) без лидирующих нулей. Если в столбце нет крестиков, то вывести 0.

#### Примеры

	<i>input.txt</i>	<i>output.txt</i>
1	1	1
2	11	11
4	110	110
6	101	101

## Задача 2. Прыжки

Ограничение по времени на 1 тест:

1 сек.

Коля играет в новую компьютерную игру под названием «Гамбург 2». Ему нужно преодолеть длинный зал с разноцветным полом.

Зал представлен отрезком  $[0, L]$  на координатной прямой. Пол раскрашен в три цвета: белый, синий и оранжевый. Нормальная скорость бега Коли равна  $u_0$ . Если Коля бежит по оранжевому полу, то он движется равноускоренно с ускорением  $a > 0$ . При беге по полу любого другого цвета: если скорость Коли больше нормальной ( $u_0$ ), то его ускорение равно  $-b < 0$ ; если его скорость равна нормальной, то его скорость не изменяется. Таким образом, бег по оранжевому полу разгоняет Колю, а бег по полу другого цвета постепенно уменьшает его скорость до нормальной. Находясь на синем полу, Коля может прыгнуть. При этом он поднимается в воздух на произвольное время, не превосходящее  $T_0$ . Пока Коля летит по воздуху, находящийся под ним пол никак на него не действует. Скорость в полёте, при прыжке и при приземлении не изменяется.

Раскраска пола в зале задаётся следующим образом. Отрезок  $[0, L]$  разбит на отрезки-участки. На каждом таком участке пол имеет один цвет. По неизвестной Коле причине после каждого участка оранжевого цвета следующий участок имеет синий цвет. На концах отрезков действуют два цвета. Например, если кто-то стоит на любом конце отрезка с синим цветом, то он может прыгать.

Начальная скорость Коли равна нормальной ( $u_0$ ). Определите минимально возможное время, за которое Коля может преодолеть зал.

### Входные данные

В первой строке входного файла записано пять целых чисел  $L, u_0, a, b, T_0$  ( $1 \leq u_0, a, b, T_0 \leq 100, 1 \leq L \leq 10^9$ ). Во второй строке дано количество участков  $N$  в зале ( $1 \leq N \leq 100000$ ). Следующая строка содержит числа  $x_i$  ( $0 \leq i \leq N$ ) – координаты границ участков ( $0 \leq x_i \leq L, x_0 = 0, x_N = L$ ). Последовательность  $x_i$  содержит  $N+1$  число и строго возрастает. В последней строке записана последовательность, состоящая из  $N$  символов, определяющая цвета участков. Буква “W” обозначает белый цвет, “B” – синий, а “O” – оранжевый. Гарантируется, что после каждого символа “O” сразу идёт символ “B”.

### Выходные данные

В выходной файл необходимо вывести одно вещественное число – минимальное время, за которое можно преодолеть зал. Ответ считается верным в том случае, если абсолютная или относительная погрешность не превосходит  $10^{-8}$ .

### Пример

<i>input.txt</i>	<i>output.txt</i>
50 10 2 2 1 3 0 11 17 50 OBW	4.5

### Комментарий

Оптимальный пробег теста из условия устроен так:

- Начальное состояние:  $t = 0, x = 0, V = 10$ .
- Пробегаем оранжевый участок ( $t = 1, x = 11, V = 12$ ).
- На границе с синим участком прыгаем, поднимаясь в воздух на полсекунды ( $t = 1.5, x = 17, V = 12$ ).
- Оказавшись на конце синего участка, совершаем прыжок на секунду ( $t = 2.5, x = 29, V = 12$ ).
- Приземлившись на белый участок, бежим по нему одну секунду с замедлением ( $t = 3.5, x = 40, V = 10$ ).
- Скорость упала до нормальной, бежим ещё одну секунду с постоянной скоростью ( $t = 4.5, x = 50, V = 10$ ).

### Задача 3. Пробки

Ограничение по времени на 1 тест: **1 сек.**

К 2040 году пробок стало ещё больше. Многие дороги реконструировали, но все-таки остались узкие участки дороги. По таким участкам машины могут двигаться только в одну сторону и не могут обгонять друг друга. И по правилам интервал между машинами во время движения не должен быть меньше заданного для этого участка.

Перед началом участка организована площадка со светофором. На этой площадке машины могут выстроиться в очередь в любом порядке. После того как загорится зеленый сигнал светофора, машины начинают движение по этому участку. При этом следующая машина начинает движение сразу после того как предыдущая отъехала на расстояние разрешенного интервала. Считается, что машины набирают свою максимальную скорость моментально.

Помогите определить минимальное время, необходимое для проезда всех машин.

#### Входные данные

В первой строке дано три целых числа  $N$  — количество машин, скопившихся перед светофором,  $S$  — длина узкого участка в метрах,  $D$  — минимальный разрешенный на этом участке интервал ( $1 \leq N \leq 500000$ ,  $1 \leq D \leq S \leq 10^8$ ). Далее идет  $N$  строк, каждая из которых содержит описание одной машины, которое состоит из двух целых чисел  $V_i$  — скорость машины в м/с, и  $L_i$  — длина машины в метрах ( $1 \leq V_i \leq 100$ ,  $1 \leq L_i \leq 100$ ).

#### Выходные данные

В выходной файл необходимо вывести одно число — минимальное время с точностью  $10^{-5}$ , за которое все машины проедут заданный участок, соблюдая все правила.

#### Примеры

<i>input.txt</i>	<i>output.txt</i>
2 100 10 10 5 5 10	23.50000

## Задача 4. Лампочки

Ограничение по времени на 1 тест: **1 сек.**

Электронное табло на стадионе «Газ-Мяс» представляет собой таблицу из лампочек размером  $M \times N$ . К несчастью, во время последнего матча пульт управления серьёзно испортился. Теперь им можно только изменять состояние лампочек на противоположное в любом ряду (в строке или в столбце). Директор стадиона просит вас вывести на экран рекламный баннер. Вам необходимо определить, можно ли это сделать. Изначально табло выключено.

### Входные данные

В первой строке входного файла записано два натуральных числа  $M$  и  $N$  ( $1 \leq M, N \leq 10^3$ ), далее следует описание рекламного баннера. Оно представляет собой  $M$  строк из  $N$  символов. Символ «1» означает, что соответствующая лампочка должна быть включена, а символ «0» — должна быть выключена.

### Выходные данные

В единственной строке выходного файла должно быть записано одно слово: **YES**, если баннер вывести на табло возможно, и **NO** — иначе.

### Примеры

<i>input.txt</i>	<i>output.txt</i>
2 2 11 01	<b>NO</b>
3 3 010 101 010	<b>YES</b>

## Задача 5. Измерения

Ограничение по времени на 1 тест:

4 сек.

Дано  $N$  узлов на электрической схеме. Требуется узнать напряжения между каждой парой узлов. Потенциал в  $i$ -ом узле равен  $\varphi_i$ . Тогда напряжение между  $i$ -ым и  $k$ -ым узлами равно разности потенциалов в узлах, т.е.  $U_{ik} = \varphi_k - \varphi_i$ .

Изначально все потенциалы и напряжения неизвестны. Производится серия измерений напряжения. В результате измерения между узлами  $i$  и  $k$  становится известным напряжение между ними  $U_{ik}$ . Величина напряжения определяется абсолютно точно. Знак напряжения также становится известным.

Есть опасность, что какое-то из измерений было произведено неправильно. При неправильном измерении величина напряжения определяется неправильно. Однако неизвестно, какие именно напряжения было измерены неверно.

Задан порядок проведения замеров — это последовательность пар узлов, между которыми производятся замеры напряжения. Найдите:

1. После какого измерения можно будет точно определить все напряжения в схеме, если предположить, что все измерения правильные.
2. После какого измерения можно будет гарантированно определить все напряжения, если предположить, что неправильных измерений не более одного.

### Входные данные

В первой строке входного файла дано два целых числа  $N$  — количество интересующих нас узлов и  $M$  — количество измерений в плане ( $2 \leq N \leq 1000$ ,  $1 \leq M \leq 3000$ ). Далее идёт  $M$  строк, каждая из которых содержит по два целых числа  $a_i$  и  $b_i$  — номера узлов, участвующих в  $i$ -ом измерении ( $1 \leq a_i \neq b_i \leq N$ ).

### Выходные данные

В выходной файл необходимо выдать два целых числа, первое из них — это номер первого измерения, после которого можно однозначно определить все напряжения, при условии, что нет ошибок в измерениях. Второе — номер первого измерения, после которого все напряжения можно определить, считая, что не более одного измерения проведено неверно.

Если какого-то номера не существует, то выведите вместо него  $-1$ .

### Примеры

<i>input.txt</i>	<i>output.txt</i>
2 5 1 2 2 1 1 2 2 1 1 2	1 3
3 1 1 2	-1 -1
5 7 1 5 5 4 4 3 3 1 5 2 2 3 2 1	5 -1

## Задача 6. Спасение отца Федора

Ограничение по времени на 1 тест: **1 сек.**

Отец Федор идет пешком из Тифлиса по Военно-Грузинской дороге, а навстречу ему двигаются Остап с Кисой. После недолгой стычки с особой, приближенной к императору, как мы знаем, служитель культа в панике забирается на самую высокую скалу, находящуюся на расстоянии не более  $R$  от места битвы с титаном мысли. Снимать же его придется уже пожарной команде.

Возникает закономерный вопрос, какой высоты лестницу им надо с собой захватить? При этом спасатели не знают точного места стычки, но полагают, что это могло быть в одной из известных им мест. Им надо знать два числа — *минимально допустимую* и *достаточную* длины лестницы. *Минимально допустимой* длины, может быть, и хватит, чтобы снять похитителя колбасы, но уж меньшей-то точно не хватит. А *достаточная* длина — это минимальная из всех возможных длин лестницы, с помощью которой они наверняка снимут отца Федора.

### Входные данные

В первой строке входного файла записано через пробел три целых числа  $N$ ,  $k$  и  $R$  ( $1 \leq N \leq 100000$ ,  $1 \leq k \leq 1000 \leq N$ ,  $0 \leq R \leq 1000$ ).  $N$  — длина Военно-Грузинской дороги,  $k$  — количество возможных мест столкновения,  $R$  — то расстояние от места стычки, в пределах которого отец Федор подыскивает самую высокую скалу. Во второй строке приведены  $N$  чисел  $H_i$ , описывающих скалы на Военно-Грузинской дороге ( $0 \leq H_i \leq 1000$ ,  $H_i$  — высота скалы в точке  $i$ ,  $1 \leq i \leq N$ ). В третьей строке приведены  $k$  чисел  $l_i$  — возможные места столкновения ( $1 \leq l_i \leq N$ ).

### Выходные данные

В выходной файл необходимо вывести два числа — минимально допустимую и достаточную длины лестниц.

### Пример

<i>input.txt</i>	<i>output.txt</i>
20 3 2 2 3 5 6 9 20 1 6 13 45 6 12 9 3 0 2 4 5 3 2 2 16 12	5 45

## Задача 7. Сокровища

Ограничение по времени на 1 тест: 2 сек.  
Java 3 сек.

Погоня за сокровищами — дело не только утомительное, но и иногда чреватое неприятными сюрпризами. Карабас-Барабас бежит за Золотым ключиком, а Остап Бендер — за бриллиантами. Но если твой шарф намертво зацепился за дерево, как борода у Карабаса-Барабаса, то тут далеко не убежишь. Более того, до весьма болезненного столкновения с сосной осталось совсем немного. Определите, сколько метров осталось бежать до тех пор, пока не раздастся громкий "Буммм", который и обозначит это столкновение.

### Входные данные

В первой строке входного файла записано одно целое число  $n$  — количество наборов тестов ( $1 \leq n \leq 10^5$ ). В следующих  $n$  строках даны тестовые наборы, по одному на каждой строке. Каждый тестовый набор состоит из восьми целых чисел, записанных через пробел  $x, y, r, x_0, y_0, x_1, y_1, d$ .

$x, y$  и  $r$  — координаты центра сосны и ее радиус ( $-10^4 \leq x, y \leq 10^4, 1 \leq r \leq 10^4$ ).

$x_0, y_0$  — координаты точки на сосне, к которой приклеен шарф ( $-10^4 \leq x_0, y_0 \leq 10^4$ ).

$x_1, y_1$  — координаты самого великого комбинатора ( $-10^4 \leq x_1, y_1 \leq 10^4$ ). При этом длина его шарфа равна расстоянию между этими двумя точками, и шарф не проходит сквозь сосну.

$d$  — направление движения: 1, если бег происходит по часовой стрелке и  $-1$ , если против нее. Остап бежит в заданном направлении так, что шарф всегда натянут и при беге наматывается на сосну.

### Выходные данные

В выходной файл необходимо вывести одно вещественное число — длину траектории бега великого комбинатора до полного трагедии конца — с абсолютной либо относительной точностью не менее  $10^{-6}$ .

### Примеры

<i>input.txt</i>	<i>output.txt</i>
1 0 0 1 1 0 2 3 -1	6.017464590

## Задача 8. Фигура Безье

Ограничение по времени на 1 тест:

1 сек.

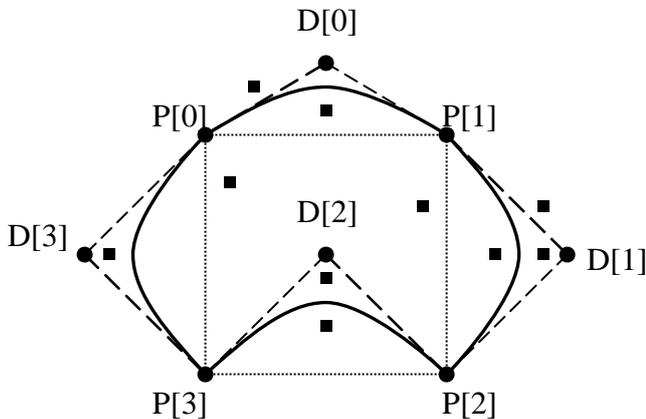
Дан выпуклый многоугольник, заданный набором точек  $P[0] \dots P[n]$ . Точки заданы в порядке обхода по часовой стрелке. Для удобства первая и последние точки совпадают  $P[0] = P[n]$ . Также задан набор точек  $D[0] \dots D[n-1]$ . Каждая упорядоченная тройка точек  $P[i], D[i], P[i+1]$  задает квадратичную кривую Безье, параметрическую кривую, состоящую из множества точек

$$B[i](t) = P[i] \cdot (1-t)^2 + 2 \cdot t \cdot (1-t) \cdot D[i] + P[i+1] \cdot t^2,$$

где  $0 \leq t \leq 1$ .

Известно, что эти кривые не пересекаются друг с другом.

Есть набор точек  $S[1] \dots S[m]$ . Требуется определить, какие из этих точек лежат внутри фигуры образованной описанными кривыми. Если точка лежит на границе, считать что она находится внутри фигуры.



### Входные данные

В первой строке входного файла записаны через пробел два числа  $n$  и  $m$  ( $4 \leq n < 10^3$ ,  $1 \leq m < 10^4$ ). Далее записаны  $n$  координат точек многоугольника ( $P[i]$ ), затем  $n$  координат точек  $D[i]$ , задающих кривые Безье. После них записано  $m$  координат точек, для которых нужно определить их место расположения. Координаты всех точек заданы целыми числами, по модулю не превосходящими  $10^8$ .

### Выходные данные

В выходной файл необходимо вывести количество точек, находящихся внутри фигуры.

### Пример

<i>input.txt</i>	<i>output.txt</i>
<pre> 4 10 0 10 10 10 10 0 0 0 5 13 15 5 5 5 -5 5 1 8 5 11 9 7 -4 5 12 5 14 5 5 2 5 4 14 7 2 12                     </pre>	<pre> 5                     </pre>

## Задача 9. Числа

Ограничение по времени на 1 тест: **1 сек.**

На бухгалтера Берлагу пребывание в сумасшедшем доме подействовало очень сильно. С тех пор он только тем и занимается, что отсчитывает на счетах все числа подряд — занятие, слава богу, советской властью не запрещенное. И если хотя бы одна цифра числа равна семи, то он это число аккуратнейшим образом в тетрадочку записывает.

Но чисел таких очень много, а любимый журнал "Ярбух фюр психоаналитик унд психопатологик" почитать очень хочется. Помогите бедному бухгалтеру, посчитайте за него, сколько чисел в диапазоне от  $a$  до  $b$  включительно в десятичной записи содержит хотя бы одну семерку.

### Входные данные

Во входном файле через пробел записано два целых неотрицательных числа  $a$  и  $b$  ( $0 \leq a \leq b \leq 10^{18}$ ).

### Выходные данные

В выходной файл необходимо вывести одно целое число — количество чисел на заданном интервале, в десятичной записи которых встречается хотя бы одна цифра 7.

### Пример

	<i>input.txt</i>	<i>output.txt</i>
1	100	19

## Задача 10. Космический корабль

Ограничение по времени на 1 тест:

2 сек.

Чтобы вовремя успеть на олимпиаду в Сочи, Зойчу необходимо вылетать с его родной планеты уже сейчас. На приборной панели его космического корабля есть несколько кнопок. У каждой кнопки есть четыре состояния, пронумерованных числами от 0 до 3. Каждую кнопку вручную можно переводить из текущего состояния в состояние на единицу больше, либо на единицу меньше. Для того чтобы уменьшить состояние нужно затратить одну единицу энергии, а чтобы увеличить – две.

Но кнопки на этом корабле не простые: ручное изменение состояния каждой из них может привести к изменению состояния других. Например, при ручном увеличении состояния первой кнопки, для третьей и пятой оно тоже увеличится на одну единицу, а для второй уменьшится на две единицы. Больше максимального значения состояние подняться не может, и состояние опуститься ниже минимального также не может. Такие правила заданы по два для каждой кнопки: одно для увеличения, другое для уменьшения состояния. При этом энергия затрачивается только на те кнопки, состояния которых изменяются вручную, автоматические изменения состояний происходят без энергозатрат. Ручное увеличение состояния 3 и уменьшение состояния 0 кнопки может привести к поломке, поэтому такие действия запрещены. Если даже вы попытаетесь, то состояния кнопок не изменятся, правила применяться не будут, энергия не уменьшится.

Зойчу нужно перевести кнопки из начального состояния в состояние, необходимое для взлёта. Помогите ему сделать это с минимальными затратами энергии.

### Входные данные

В первой строке входного файла содержится единственное целое число  $N$  — количество кнопок на приборной панели космического корабля ( $1 \leq N \leq 9$ ).

Во второй строке задано начальное состояние кнопок, в третьей — состояние, необходимое для взлёта. Каждое состояние представляет собой последовательность из  $N$  целых чисел  $s_i$  ( $0 \leq s_i \leq 3$ ), где  $s_i$  соответствует состоянию  $i$ -й кнопки.

Далее для каждой кнопки в порядке возрастания номеров заданы два правила, каждое на отдельной строке. Первое соответствует увеличению состояния кнопки, второе — уменьшению. Каждое правило представляет собой последовательность из  $N$  целых чисел  $r_i$  ( $-3 \leq r_i \leq 3$ ) и задает величину, на которую увеличивается состояние  $i$ -й кнопки при применении данного правила. В правилах для каждой кнопки  $r_k = 0$ , если эта кнопка имеет номер  $k$  ( $1 \leq k \leq N$ ).

### Выходные данные

В выходной файл необходимо вывести одно целое число — минимальное количество энергии, которое необходимо затратить, чтобы перевести кнопки на приборной панели космического корабля из начального состояния в состояние, необходимое для взлёта. Вывести  $-1$ , если такое состояние получить нельзя.

**Примеры**

<i>input.txt</i>	<i>output.txt</i>
3 0 0 0 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0	12
3 3 3 3 3 2 3 0 0 0 0 0 0 0 0 0 0 0 -1 0 1 0 0 0 0	-1