

## Задача А. Вычислительная агрономия

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

Король Квадратии планирует провести аграрную реформу. Квадратия представляет собой прямоугольник, состоящий из  $m \times n$  квадратиков. Квадратики задаются парами чисел  $(x, y)$ , где  $x$  изменяется от 1 до  $m$ , а  $y$  — от 1 до  $n$ . На каждом квадратике может быть расположен либо дом крестьянина, либо болото, либо поле. Король хочет выделить каждому крестьянину поле таким образом, чтобы каждому крестьянину было выделено ровно одно поле, и на каждое поле было выделено не более чем одному крестьянину.

Король попросил своего министра агрономии подготовить список всех крестьян королевства. После этого он назначит им поля. Тайный советник короля вызнал, по какому алгоритму король будет назначать крестьянам поля.

Король по очереди будет рассматривать крестьян в списке. Для каждого крестьянина он выберет поле, которое находится на минимальном расстоянии от дома этого крестьянина. Если таких полей несколько, то будет выбрано поле с минимальным  $x$ . Если полей по прежнему несколько, будет выбрано поле с минимальным  $y$ . В качестве меры расстояния от крестьянина до поля король планирует использовать так называемое «манхаттанское расстояние», расстояние между квадратиками с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  равно  $|x_1 - x_2| + |y_1 - y_2|$ .

Министр агрономии хотел бы упорядочить крестьян в списке таким образом, чтобы сумма расстояний от дома крестьянина до поля, которое ему выделено была как можно меньше. Помогите ему найти такой порядок.

### Формат входного файла

Первая строка входного файла содержит четыре целых числа:  $m$ ,  $n$ ,  $k$  и  $s$  — размеры страны, количество крестьян и количество болот, соответственно ( $1 \leq m, n \leq 20$ ,  $1 \leq k \leq mn/2$ ,  $0 \leq s \leq mn - 2k$ ). Следующие  $k$  строк содержат координат домов, где живут крестьяне. Никакие два крестьянина не живут на одном и том же квадратике.

Следующие  $s$  строк содержат координаты квадратиков, на которых расположены болота.

### Формат выходного файла

Выведите  $k$  номеров — в каком порядке министру агрономии следует упорядочить крестьян, чтобы король выделил им поля оптимальным образом.

### Пример

agrarian.in	agrarian.out
3 5 5 0	3 4 2 1 5
2 3	
2 4	
1 3	
2 2	
3 3	

## Задача В. Вычислительная биология

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

В современной биологии ученым часто приходится иметь дело с последовательностями ДНК. Эти последовательности зачастую являются очень длинными и их ручная обработка требует большого количества времени и сил. Поэтому возникает идея автоматизировать этот процесс.

Для этого можно применять компьютерные методы обработки данных, например, весьма полезными оказываются алгоритмы на строках. В этой задаче последовательность ДНК будет представляться в виде строки, все символы которой входят в множество  $\{A, G, C, T\}$ .

Пусть даны две последовательности ДНК:  $s = s_1s_2\dots s_n$  и  $t = t_1t_2\dots t_m$ . Будем говорить, что  $t$  может получиться *в результате эволюции* из  $s$ , если  $s$  является подпоследовательностью  $t$ , то есть существует такая последовательность индексов  $1 \leq i_1 < i_2 < \dots < i_m \leq n$ , что  $s_1 = t_{i_1}, s_2 = t_{i_2}, \dots, s_m = t_{i_m}$ . Необходимо выяснить, может ли последовательность  $t$  получиться в результате эволюции из  $s$ .

### Формат входного файла

Первая строка входного файла содержит последовательность  $s$ , вторая — последовательность  $t$ . Размер входного файла не превосходит 256 килобайт.

### Формат выходного файла

В выходной файл выведите слово YES, если последовательность  $t$  могла получиться в результате эволюции из  $s$ , и слово NO — иначе.

### Примеры

<code>biology.in</code>	<code>biology.out</code>
GTA AGCTA	YES
AAAG GAAAAAT	NO

## Задача С. Вычислительная комбинаторика

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

Напомним, что называется правильной скобочной последовательностью:

- пустая строка является правильной скобочной последовательностью;
- если строка  $a$  — правильная скобочная последовательность, то строки  $(a)$ ,  $[a]$  — тоже правильные скобочные последовательности;
- если строки  $a$  и  $b$  — правильные скобочные последовательности, то строка  $ab$  — тоже правильная скобочная последовательность.

Задана строка  $S$ , состоящая из квадратных и круглых скобок. Разрешается заменять квадратную открывающую скобку (`[`) на круглую открывающую (`(`) и наоборот, а также квадратную закрывающую скобку (`]`) на круглую закрывающую (`)`) и наоборот.

За одно действие разрешается изменить ровно один символ строки. Необходимо за минимальное число действий преобразовать  $S$  в правильную скобочную последовательность.

### Формат входного файла

Входной файл содержит строку  $S$ . Ее длина не превосходит 100 000 символов.

### Формат выходного файла

В выходной файл выведите искомое минимальное число действий, или  $-1$ , если преобразовать  $S$  в правильную скобочную последовательность невозможно.

### Примеры

<code>brackets.in</code>	<code>brackets.out</code>
<code>(( )) []</code>	0
<code>[ ( ]</code>	2
<code>(( [ ] ])</code>	-1

## Задача D. Вычислительная геометрия

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

Подземный пешеходный переход на станции метро “Лесная” начали строить одновременно с туннелем под Ла-Маншем. Так вот — последний построили быстрее.

---

с форума, посвященного дорогам Санкт-Петербурга

Как-то на Васильевском ко мне подошли старушки-активистки и предложили расписаться против строительства Западного скоростного диаметра. Когда я спросил, где можно расписаться “за” Западный скоростной диаметр — чуть кошелкой по голове не получил...

---

с форума, посвященного дорогам Санкт-Петербурга

*Открытие второй очереди Вантового моста и второй эстакады в Мурино на КАД Санкт-Петербурга посвящается...*

Правительство одного города решило построить вокруг города кольцевую автодорогу. Однако, к сожалению, фонды, выделенные на строительство были быстро... кончились и только часть дороги была построена. Через несколько лет решение построить дорогу снова назрело в умах руководителей города и решено было продолжить строительство. Однако развитие города привело к тому, что старый проект решено было забросить и строить новую дорогу. Но и на этот раз фондов на всех не хватило... в смысле, не хватило на строительство дороги, поэтому дорогу достроить снова не удалось.

И вот, правительство снова исследует возможность строительства кольцевой дороги! Перед тем, как проектировать новую дорогу, решено было изучить, что же настроили в прошлые разы. Каждый из построенных фрагментов дороги представляет собой дугу окружности. Для начало решено было выяснить, каково минимальное расстояние между построенным фрагментами дороги.

### Формат входного файла

Входной файл содержит две строки, каждая строка описывает одну из построенных частей дороги. Каждая часть дороги описывается пятью целыми числами:  $x$ ,  $y$ ,  $r$ ,  $b$  и  $e$  — координаты центра и радиус запланированной кольцевой автодороги и начальный и конечный углы, между которыми ее удалось таки построить. Углы заданы в градусах и отсчитываются против часовой стрелки от восточного направления (ось  $Oy$  направлена на север, а ось  $Ox$  — на восток), если  $e < b$ , это означает, что часть дороги проходит через восточное направление. Все координаты не превышают 1000 по абсолютной величине, радиусы положительны и не превышают 1000,  $b$  и  $e$  изменяются от 0 до 359, включительно,  $b \neq e$ .

### Формат выходного файла

Выведите одно вещественное число — кратчайшее расстояние между фрагментами дорог. Ваш ответ должен отличаться от правильного не более чем на  $10^{-5}$ .

### Пример

<code>circular.in</code>	<code>circular.out</code>
0 0 4 270 90 6 0 1 90 240	1.0
0 0 4 270 90 5 0 2 90 240	0.0

## Задача Е. Вычислительная криптография

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

Профессор Умник работает над взломом нового шифра — Формального Индифферентного Гармонического Виртуального Алгоритма Маскирования (ФИГВАМ). После некоторых исследований профессор пришел к выводу, что для восстановления ключа шифрования следует решить следующую задачу: найти два числа  $x$  и  $y$  ( $1 \leq x, y \leq n$ ), такие что:

- $x \wedge y = y$ ;
- $(ax + by) \oplus (ay + bx)$  максимально.

Здесь  $p \wedge q$  означает побитовое “и” чисел  $p$  и  $q$ , а  $p \oplus q$  означает побитовое “исключающее или” чисел  $p$  и  $q$  (операция `xor` в паскале).

Вам заданы числа  $n$ ,  $a$  и  $b$ . Помогите профессору найти  $x$  и  $y$ .

### Формат входного файла

Входной файл содержит  $n$ ,  $a$  и  $b$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq a, b \leq 2000$ ).

### Формат выходного файла

Выведите  $x$  и  $y$ , удовлетворяющие условию. Если решений несколько, выведите любое.

### Пример

<code>crypto.in</code>	<code>crypto.out</code>
20 2 3	15 10

## Задача F. Вычислительная теория игр

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

Петя играет в интересную игру. Для этой игры необходима монетка. Петя подбрасывает ее  $n$  раз и считает, сколько раз выпадает «решка». Если решка выпадает хотя бы  $m$  раз, то Петя считает, что он выиграл игру.

Однажды Петя задумался, какова вероятность того, что он выиграет игру. Для этого он хочет найти количество последовательностей результатов подбрасывания монетки, содержащих ровно  $n$  подбрасываний, при которых «решка» выпала хотя бы  $m$  раз.

Помогите Пете — найдите это число, считая, что при каждом броске монетка может выпасть либо «орлом», либо «решкой».

### Формат входного файла

Входной файл содержит два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 20$ ,  $0 \leq m \leq n$ ).

### Формат выходного файла

В выходной файл выведите ответ на задачу.

### Примеры

<code>game.in</code>	<code>game.out</code>
2 0	4
3 2	4

## Задача G. Вычислительная генеалогия

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

В современной биологии ученым часто приходится иметь дело с последовательностями ДНК. Эти последовательности зачастую являются очень длинными и их ручная обработка требует большого количества времени и сил. Поэтому возникает идея автоматизировать этот процесс.

Для этого можно применять компьютерные методы обработки данных, например, весьма полезными оказываются алгоритмы на строках. В этой задаче последовательность ДНК будет представляться в виде строки, все символы которой входят в множество  $\{A, G, C, T\}$ .

Пусть даны две последовательности ДНК:  $s = s_1s_2 \dots s_n$  и  $t = t_1t_2 \dots t_m$ . Будем говорить, что  $t$  может получиться *в результате эволюции* из  $s$ , если  $s$  является подпоследовательностью  $t$ , то есть существует такая последовательность индексов  $1 \leq i_1 < i_2 < \dots < i_m \leq n$ , что  $s_1 = t_{i_1}$ ,  $s_2 = t_{i_2}$ , ...,  $s_m = t_{i_m}$ .

С целью исследования генеалогии организма необходимо выяснить, сколько различных последовательностей  $s$ , таких что заданная последовательность  $t$  могла получиться в результате эволюции из  $s$ . Выведите ответ по модулю  $10^9 + 7$ .

### Формат входного файла

Первая строка входного файла содержит последовательность  $t$ . Размер входного файла не превосходит 256 килобайт.

### Формат выходного файла

В выходной файл одно число — количество различных подпоследовательностей последовательности  $t$  по модулю  $10^9 + 7$ .

### Примеры

<code>genealogy.in</code>	<code>genealogy.out</code>
ATGA	15

Возможные подпоследовательности: «», «A», «AA», «AG», «AGA», «AT», «ATA», «ATG», «ATGA», «G», «GA», «T», «TA», «TG», «TGA».

## Задача Н. Вычислительная лингвистика

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

Руны — это древние магические знаки, которые наши предки использовали как буквы. Говорят, что рунные знаки обладают магическими свойствами, а при сложении рун в слова их магическая сила многократно возрастает. Если кузнец изготовит доспехи и начертит там определенные руны в определенном порядке, то доспехи будут наделены необычайными магическими силами.

Для того, чтобы стать обладателем таких доспехов достаточно просто принести кузнецу начертания этих рунных знаков. А вот, чтобы стать обладателем рунного знака приходилось немало потрудиться. Воины добывали начертания рун других языков и наречий в боях или получали их в качестве наград в благодарность за оказанные услуги.

Но так или иначе и в этом деле развелись жулики. По подозрениям ученых кузнец Игнатус Мошеникус изготавливал благородным войнам фальшивые рунные слова. Из древних преданий ученым стало достоверно известно, что каждая руна записывается из двух, трех или четырех латинских букв. Причем первая буква рунного слова всегда записывается как заглавная, а все остальные являются маленькими. Ученые перевели несколько, выкованных этим кузнецом, рунных слов на латинский язык и теперь нуждаются в Вашей помощи. Проверьте, является ли приведенное слово рунным.

### Формат входного файла

В единственной строке входного файла содержится слово. Оно представляет собой непустую строку, длиной не более 100000 символов, содержащую только большие и маленькие буквы латинского алфавита.

### Формат выходного файла

Если введенное слово является рунным, запишите в выходной файл слово «Yes», в противном случае — «No»

### Примеры

<code>runes.in</code>	<code>runes.out</code>
<code>IoTheBest</code>	<code>Yes</code>
<code>IoItIsWaste</code>	<code>No</code>



## Задача I. Вычислительная теория вероятностей

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

Петя собирается принять участие в новом телешоу. В процессе шоу игрок может заработать деньги. Исходно у игрока имеется 100 рублей. После этого игроку последовательно задаются  $n$  вопросов, один за другим. Если игрок отвечает на вопрос правильно, сумма, которой он располагает, удваивается. Если же игрок отвечает неправильно, то он ничего не получает и покидает телешоу.

Перед каждым вопросом (в том числе первым) игрок может покинуть шоу, забрав те деньги, которые у него есть на данный момент. Кроме того, один раз за игру игрок может купить «страховку». Страховка стоит  $c$  рублей, которые вычитаются из суммы игрока перед вопросом. Если купить перед некоторым вопросом страховку, то в случае неправильного ответа сумма, которой располагает игрок не удваивается, но игрок продолжает игру. Если же он отвечает правильно, то как обычно, сумма удваивается. Чтобы купить страховку игрок должен иметь строго больше  $c$  рублей.

Друг Пети работает на телевидении и ему удалось добыть список тем, на которые Пете будут заданы вопросы. К сожалению, сами вопросы добыть не удалось. Теперь для каждого вопроса Петя знает число  $p_i$  — вероятность того, что он правильно ответит на этот вопрос. Петя хочет разработать оптимальную стратегию игры, которая позволит ему максимизировать ожидаемый выигрыш. Помогите ему.

### Формат входного файла

Первая строка входного файла содержит два целых числа  $n$  и  $c$  ( $1 \leq n \leq 50$ ,  $1 \leq c \leq 10^9$ ). Вторая строка содержит  $n$  целых чисел в диапазоне от 0 до 100 — вероятности что Петя ответит на соответствующий вопрос правильно, в процентах.

### Формат выходного файла

Выведите одно вещественное число — ожидаемый приз Пети, если он будет следовать оптимальной стратегии. Ответ должен отличаться от правильного не более чем на  $10^{-8}$ .

### Пример

<code>tvshow.in</code>	<code>tvshow.out</code>
2 100 50 50	100
2 50 50 50	112.5
2 50 60 0	120

Во втором примере оптимальной стратегией будет купить страховку перед вторым вопросом. В таком случае ожидаемый приз будет  $1/2 \times 0 + 1/2 \times (1/2 \times 150 + 1/2 \times 300)$ .

В третьем примере лучше на второй вопрос даже не пытаться отвечать, а забрать деньги после первого вопроса.