

Задача 1. Японский кроссворд

Ограничение времени: 5сек

В японском кроссворде, в отличие от привычного для нас, зашифровывают не слова, а картинки. Шифрование производится следующим образом.

Шифруемой картинкой является чёрно-белое изображение на клеточном поле, например,

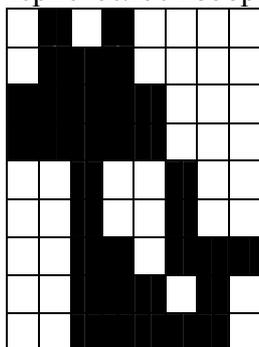


Рис. 1

Такой рисунок можно разбить на чёрные отрезки, между которыми есть хотя бы одна белая клетка, расположенные по горизонтали и вертикали. В качестве шифра указываются длины чёрных отрезков, расположенных в каждой строке и каждом столбце клеточного поля.

Например, картинку, представленную на рис. 1, можно представить следующим образом:

1 строка 1,1	1 столбец 2
2 строка 3	2 столбец 4
3 строка 5	3 столбец 8
4 строка 5	4 столбец 4,3
5 строка 1,1	5 столбец 2,2
6 строка 1,1	6 столбец 3,1
7 строка 2,3	7 столбец 3
8 строка 3,1	8 столбец 1
9 строка 5	

Если в строках или столбцах клеточного поля несколько чёрных отрезков, то длины этих отрезков указываются в строках слева направо, а в столбцах сверху вниз.

Заметим, что для рассмотренного метода шифрования картинка дешифровка не всегда является однозначной.

Будем считать правильно составленным кроссвордом такой кроссворд, для которого гарантируется существование хотя бы одного решения. Необходимо решить заданный правильно составленный японский кроссворд. При этом, если решение не единственное, то вывести любое из них.

Входные данные (input.txt)

В первой строке входного файла содержится два целых числа: N и M , где N – количество строк в клеточном поле ($1 \leq N \leq 15$), M – количество столбцов ($1 \leq M \leq 15$). Далее следует N строк в следующем формате: $k \ l_1 \ l_2 \ \dots \ l_k$, где k – число отрезков, l_i – длина i -го отрезка. Далее следует M строк в таком же формате, содержащих информацию о столбцах рисунка.

Выходные данные (output.txt)

В выходной файл вывести восстановленную картинку, обозначая чёрную клетку символом '*', а белую клетку символом '.', без пробелов. Каждая строчка выходного файла должна содержать одну строку картинки.

Пример**input.txt**

9 8
2 1 1
1 3
1 5
1 5
2 1 1
2 1 1
2 2 3
2 3 1
1 5
1 2
1 4
1 8
2 4 3
2 2 2
2 3 1
1 3
1 1

output.txt

```
. * . * . . . .  
. * * * . . . .  
* * * * * . . .  
* * * * * . . .  
. . * . . * . .  
. . * . . * . .  
. . * * . * * *  
. . * * * . * .  
. . * * * * * .
```

Задача 2. Уфология

Ограничение времени: 5сек

Школьник Вася должен был участвовать в военно-полевой игре «Зарница 2008». Жюри определило правила проведения игры:

- 1) в лесу имеется N полянок и M тропинок между ними;
- 2) каждая тропинка соединяет две различные полянки;
- 3) тропинку можно пробегать как в прямом, так и в обратном направлении;
- 4) бегать можно только по тропинкам;

Вася очень хочет победить, и поэтому решил тренироваться. Тренер поставил перед Васей ряд задач: во-первых, пробегать каждую тропинку как можно быстрее, во-вторых, найти самый быстрый путь от I полянки к J полянке. В результате тренировок Вася знал все самые короткие пути между любыми двумя полянками. Засекая время, тренер обнаружил, что Вася пробегает любую тропинку в прямом и в обратном направлении за одинаковое постоянное время.

Когда до проведения игры оставалось несколько дней, в районе проведения игр пролетел неопознанный летательный объект (НЛО) и, по свидетельству очевидцев, приземлился на одной из полянок в том лесу, в котором должна состояться «Зарница 2008». Это событие не отменило Васину тренировку, которая состоялась на следующий день после полёта НЛО. По плану тренировки Вася должен был стартовать на полянке с номером S , а финишировать на полянке с номером F . Вася бежал по оптимальному маршруту и на какой-то из полянок запнулся и упал, а когда поднялся на ноги, то оказался совсем на другой полянке. « Вот это я запнулся – «телепортнулся»,» - подумал Вася. Но времени долго думать у него не было, Вася быстро сориентировался и достиг финиша известным ему кратчайшим маршрутом. Когда он финишировал, тренер был очень удивлён, Вася пришёл намного раньше контрольного времени. Вася рассказал тренеру, что с ним случилось на маршруте. Тренер оказался уфологом-любителем и захотел изучить телепорт, но Вася никак не мог вспомнить, на какой именно полянке запнулся и где именно оказался.

Ваша задача: помочь Васе и тренеру найти те полянки, между которыми мог возникнуть телепорт. Заметим, что в угаре тренировки Вася мог забыть как то, что он телепортировался прямо со старта, так и то, что он телепортировался прямо на финиш.

Входные данные (input.txt).

Первая строка входного файла содержит числа N, M ($3 \leq N \leq 2000, 1 \leq M \leq 5000$). Следующие M строк содержат три целых числа: I, J – номера полянок, которые соединяет тропинка, и T – ($1 \leq T \leq 100$) время, за которое Вася может её пробежать.

Далее в $M+1$ строке три целых числа: S – номер полянки, с которой стартовал Вася, F – номер полянки, на которой он финишировал и t – время, за которое он прошёл этот маршрут с учётом телепорта.

Выходные данные (output.txt)

В выходной файл в каждую строку вывести два числа: K_i, L_i – номера полянок, которые могут быть соединены телепортом (телепорт работает в одну сторону). Вывести найденные пары в порядке возрастания номеров K_i , а при равных K_i – в порядке возрастания L_i .

Пример

input.txt

7 10

1 2 2

1 4 15

2 3 3

2 6 4

3 7 3

3 4 6

3 6 3

4 5 7

4 7 2

5 7 2

1 5 9

output.txt

3 4

Задача 3. Парковка

Ограничение времени: 5сек

В городе N на некоторой улице с односторонним движением, где размещено много офисов, каждое утро возникает одна и та же проблема – куда поставить автомобиль и возможно ли это вообще?

Вдоль одной из сторон улицы имеется m мест для парковки автомобилей. Места расположены в ряд и пронумерованы от 1 до m . Каждое утро n автомобилей пытается припарковаться на площади. Каждый автомобилист старается занять удобное для него место. Считая, что первоначально улица пуста, и никто со стоянки не выезжает, подсчитайте количество возможных вариантов парковки, при которых все автомобили удачно припаркуются при условии их заезда на улицу в порядке возрастания их номеров и однократного проезда по улице, и для каждого варианта определите расположение автомобилей на стоянках. Парковка удачна, если автомобиль припарковался на желаемом месте или, если к моменту приезда автомобиля желаемое место занято другим, на любом свободном месте после желаемого.

Входные данные (input.txt)

В первой строке два целых числа: n и m , $1 \leq n \leq m \leq 50$. Во второй строке n чисел, разделенных пробелом, представляющих собой желаемые номера парковок для машин в порядке возрастания их (машин) номеров.

Выходные данные (output.txt)

Вывести варианты расположения автомобилей на стоянке. В первой строке файла – число выводимых вариантов парковки, если удачная парковка невозможна, то в первую строку вывести «0». Если число вариантов превышает 10, то вывести любые 10 вариантов (в этом случае в первой строке файла должно стоять 10). Каждый вариант представить в виде строки из m чисел, разделенных пробелом, представляющих собой последовательность номеров машин в порядке возрастания номеров их парковочных мест. Если парковочное место пусто, то вместо номера машины вывести 0. После последнего числа в строке вывести символ конца строки.

Пример 1:

input.txt	output.txt
3 3	1
2 1 3	2 1 3

Пример 2:

input.txt	output.txt
3 9	0
9 9 9	

Пример 3:

input.txt	output.txt
9 9	10
3 1 4 1 5 9 2 6 5	2 4 1 3 5 7 8 9 6
	2 4 1 3 5 7 9 8 6
	2 4 1 3 5 8 7 9 6
	2 4 1 3 5 8 9 7 6
	2 7 1 3 4 5 8 9 6
	2 7 1 3 4 5 9 8 6
	2 7 1 3 4 8 5 9 6
	2 7 1 3 4 8 9 5 6
	2 7 1 3 5 4 8 9 6
	2 7 1 3 5 4 9 8 6

Задача 4. Алиса в стране чудес или коты в шляпе

Ограничение времени: 2сек

Алиса очутилась в большом-пребольшом зале. И чудеса посыпались на неё как из рога изобилия: сначала пробежал мартовский заяц, бормоча что-то о перчатках, о веере, о королеве. Потом началось безумное чаепитие. А потом прибежал умный кот в большущей шляпе, а может быть это был чеширский кот, а может быть – кот по имени Бегемот, а может быть даже и тот кот, который ходит то направо, то налево. Но всё это не важно, важно то, что кот был в шляпе, то есть дело всё в шляпе и важно то, что он решил сделать уборку в этом зале.

Но вместо того, чтобы делать уборку самому, этот кот решил, что работу сделают другие коты. Кот снял свою шляпу и из неё выскочили коты поменьше. Они все были одного роста, тоже умные и в чуть меньших шляпах, и тоже не хотели работать. Каждый из них снял шляпу, из которой в свою очередь выпрыгнули коты ещё поменьше и т. д. В конце концов, коты стали такими маленькими, что в их шляпах уже не могли поместиться другие коты. Они-то и стали убирать зал. Алиса начала считать котов, которые убирали зал, а потом подумала: «Сколько же котов ничего не делают, а сидят возле своих шляп и ждут, когда другие выполнят всю работу?». И ещё одна мысль вертелась у неё в голове: «Какой высоты будет пирамида, если все шляпы, которые были на всех котах, поставить одна на другую?»

Вы должны помочь Алисе ответить на поставленные вопросы. При этом необходимо учесть, что количество котов, сидящих в каждой шляпе, постоянно и равно N , а высота каждой шляпы является целым числом и равна $1/(N+1)$ от высоты шляпы того кота, в шляпе которого они сидели, и высота шляп работающих котов равна 1.

Входные данные(input.txt)

В единственной строке входного файла два целых положительных числа, разделённых пробелом. Первое число – это высота шляпы самого главного умного кота, а второе число – количество котов, которые убирали зал. Оба числа не превышают 10^{15} .

Выходные данные(output.txt)

В выходной файл выведите два числа через пробел: количество неработающих котов и высоту пирамиды, составленную из всех шляп, которые были надеты на котах.

Пример

input.txt

216 125

output.txt

31 671

Задача 5. Заяц

Ограничение времени: 1сек

Убегая от лисы, заяц прыгает как угорелый (хаотично прыгает) по всей своей территории, стараясь как можно больше запутать следы, и замирает под кустиком. След зайца представляет собой самопересекающуюся ломаную. Определите минимальную площадь территории зайца, рассматривая ее как выпуклый многоугольник, вершинами которого могут быть только точки, в которые прыгал заяц.

Входные данные

В первой строке одно целое число N ($N \leq 1000$) – число прыжков. Начиная со второй строки в каждой из N строк через пробел два числа: l , a , где l – длина прыжка зайца (вещественное число, $1 \leq l \leq 10$), a – направление прыжка по азимуту (целое число, $0 \leq a < 360$). Прыжки перечислены, начиная с первого и до последнего прыжка (под кустик). Азимут задается как угол, образованный направлением прыжка зайца и направлением на север (против часовой стрелки) в градусах.

Выходные данные

В первой строке одно число – минимальная площадь, округлённая до целых.

Пример

input.txt	output.txt
3	4
2.828 225	
2.000 90	
2.828 315	

Задача 6. Систематизация

Ограничение времени: 1сек

Профессор-филолог N , выдавая задание своему студенту на проработку литературного источника, выделил символами “/*” и “*/” фрагменты текста, которые надо проработать тщательнейшим образом: символами “/*” он пометил начало фрагмента, а символами “*/” – конец. Для каждого фрагмента надо было составить резюме и найти ответы на поставленные вопросы. Дома, просматривая текст, студент обнаружил, что выделенные фрагменты не являются исключительно последовательными и даже бывают вложенными друг в друга и поэтому требуется большое внимание, чтобы вычленивать каждый фрагмент. Тогда он решил пронумеровать начала и концы фрагментов (при этом пары “/*” и “*/” он нумеровал подряд), но опять попал в затруднение, так как надо было сопоставить номера начал фрагментов номерам концов, и иногда студент из-за своей невнимательности просто не находил парных символов.

Помогите студенту, выдайте ему список номеров соответствующих пар начальных и конечных символов фрагментов в порядке возрастания номеров символов, задающих начало фрагмента.

Входные данные

В первой строке текст (без разделения на строки), в котором, возможно, встречаются пары символов “/*” и “*/”. Количество фрагментов не превышает 10000 и общая длина текста не превышает 10000 символов. При этом профессор делал разумные пометки, т.е. символы ‘/’ и ‘*’ в самом тексте не встречались и используются только для пометок, и пустых фрагментов профессор не помечал.

Выходные данные

В первой строке последовательность пар чисел, упорядоченная по возрастанию значения первого числа в паре.

Пример:

input.txt	
/*asdadadga/*b/*v/*ff*/kjfhkjhlksd*/fjk*/ u*/ dd/*rrr*/	
output.txt	
1 8 2 7 3 6 4 5 9 10	

Задача 7. Складская логистика

Ограничение времени: 3сек

Студент-программист Вася решил подработать на каникулах и устроился грузчиком на склад. Когда он пришёл на работу, то увидел, что склад ему достался чрезвычайно запущенный и захламленный. Фактически, свободное пространство представляло собой некоторый многоугольник. Работа Васи заключалась в том, чтобы перетаскивать получаемые товары от двери склада в указанный кладовщиком угол этого многоугольника. Новые товары ещё больше захламляли помещение, увеличивая число углов многоугольника и образуя новые коридоры и повороты. Вася устал таскать тяжёлые товары, и решил принести с собой ноутбук для расчёта кратчайшего пути перемещения каждого товара. Помогите Васе разработать модуль программы для минимизации длины пути перемещения груза.

Входные данные (input.txt)

В первой строке целое число N ($3 \leq N \leq 10000$) – число углов многоугольника. Далее в N строках по 2 целых числа через пробел заданы координаты вершин многоугольника (целые, по модулю не превышающие 10000). Вершины заданы в порядке обхода против часовой стрелки, первая вершина – дверь склада, т.е. точка, из которой начинается перемещение груза. В последней ($N+2$ -й) строке задан номер вершины (вершины нумеруются с 1), в которую необходимо переместить груз.

Выходные данные (output.txt)

В первой строке вещественное число, округлённое до 3 знаков после запятой – длина кратчайшего пути.

Пример 1

input.txt

put.txt

4
0 0
1 0
1 1
0 1
3

out

output.txt

1.414

Пример 2

input.txt

4
0 0
2 1
0 2
1 1
3

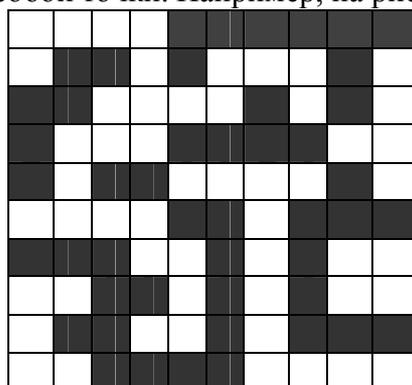
output.txt

2.828

Задача 8. Распознавание текста

Ограничение времени: 2сек

Студент Вася решил подработать и устраивается на работу в фирму, разрабатывающую системы распознавания объектов. В качестве тестового задания Васе поставили задачу разработать программу, вычисляющую количество объектов на чёрно-белом рисунке. Рисунок задан в виде прямоугольной матрицы чёрных и белых квадратных точек (пикселей). Каждый объект является одной группой чёрных точек. Отдельно стоящая чёрная точка является группой. Две чёрные точки принадлежат одной группе, если они либо являются соседними по горизонтали или вертикали, либо соседние с ними по горизонтали или вертикали точки принадлежат одной группе, т.е. группа объединяет все связанные между собой точки. Например, на рисунке



6 объектов.

Вася решил задачу. А Вы смогли бы устроиться на работу в эту фирму?

Входные данные (input.txt)

Первая строка входного файла содержит числа N, M ($1 < N \leq 1000$, $1 < M \leq 1000$). Далее построчно вводится рисунок, N строк по M символов в каждой строке. Чёрные точки обозначены символом '*' ("звёздочка"), белые – символом '.' ("точка").

Выходные данные (output.txt)

Одно целое число – число объектов.

Пример

input.txt

```
5 5
.....*
.***.
.*.*.
...*.
**.*.
```

output.txt

```
3
```

Задача 9. Лесник

Ограничение времени: 1сек

Одним из методов борьбы с лесными пожарами является метод деления участка на квадраты. Один квадрат отделяется от другого распаханными полосами. Размер таких квадратов 10 x 10 километров. Одна из обязанностей лесника объезжать свой участок, при этом он должен объехать каждый квадрат своего участка. Дом лесника расположен в юго-западной части лесного массива. Естественно, объезд он начинает от своего дома и объехав участок, возвращается домой. Если участок очень большой, то леснику требуется несколько дней для того, чтобы объехать весь участок, в этом случае лесник ночует в лесу.

Иногда, объезжая свой лесной массив, лесник должен выполнять ещё одну работу. Он должен пронумеровать все деревья, растущие по периметру квадратов. Известно, что во время объезда лесник может проехать отрезок длиной 10 км. за 10 минут или за 1 час, если во время объезда необходимо выполнить работу по разметке деревьев.

Ваша задача: определить, за какое минимальное время лесник выполнит свою работу по разметке деревьев.

Входные данные (input.txt)

Входной файл содержит два целых числа N – число квадратов по вертикали и M – число квадратов по горизонтали, разделённых пробелом ($1 \leq N, M \leq 1000$)

Выходные данные (output.txt)

В выходной файл выведите минимальное время (в минутах), за которое лесник может разметить деревья, растущие по периметру каждого квадрата лесного участка.

Пример

input.txt

1 1

output.txt

240

Задача 10. Земля – крестьянам

Ограничение времени: 5сек

В одной квадратной стране квадратный парламент принял решение о раздаче земли крестьянам. Естественно, что раздавать её предполагалось квадратными участками.

Для получения земли крестьянам всё же нужно было внести в казну небольшую символическую плату. Будем считать, что одна единица квадратной площади стоит одну «условную единицу» денег.

Крестьянин хочет потратить на приобретение земли некоторую сумму денег. Хорошо, если это число «у.е.» является полным квадратом — тогда он получит один участок. А если нет? Тогда сумму денег придётся разбить на несколько взносов и получать несколько квадратных участков. Понятно, что чем меньше число участков, тем удобнее крестьянину. Например, за 41 «у.е.» можно получить 41 квадратный участок 1x1, либо один участок 6x6 и пять участков 1x1. Возможны и другие варианты, но самым лучшим будет вариант с двумя участками: один 5x5 и один 4x4.

Необходимо разработать программу, которая подсказывает крестьянину распределение денег, минимизирующее число участков.

Входные данные (input.txt)

Во входном файле одно целое число X — сумма денег, которые предполагается потратить ($1 \leq X \leq 10\,000\,000$).

Выходные данные (output.txt)

В первой строке выходного файла — минимально возможное число участков. Во второй строке — через пробел размер стороны каждого участка в порядке убывания. Если возможно несколько решений, то во второй строке достаточно выдать любое из них.

Пример

input.txt

41

output.txt

2

5 4

Задача 11. Папуасская демократия

Ограничение времени: 2сек

Государство папуасов состоит из множества небольших островков, населённых в большей или меньшей степени. Между островками существует лодочное сообщение.

Это государство вполне демократично. С давних пор, чтобы принять какое-то глобальное решение, всё взрослое население садится в лодки и плывёт на самый большой остров (будем называть его столичным). Там проводится общее голосование, по итогам которого решение принимается или отвергается. Для принятия решения требуется строгое большинство, то есть за него должно проголосовать больше половины от общего числа голосующих.

Однако наступил момент, когда такой порядок перестал устраивать жителей столичного острова. Их вполне можно понять — голосуют-то все, а кому наводить на острове порядок после такого всеобщего сборища?!

Поэтому была предложена другая схема: сначала каждый остров голосует у себя, а потом от него на столичный остров едет один человек и сообщает там, было ли решение принято на его острове. Таким образом, голосование предлагалось сделать двухэтапным, чтобы на втором этапе решение принималось большинством островов.

Схема казалась заманчивой, однако из-за того, что острова населены по-разному, существует вероятность того, что решение будет принято, хотя его сторонники относительно численности всего населения находятся в меньшинстве.

Необходимо разработать программу, которая по количеству островов и данным об их населении определяет минимально необходимое число сторонников решения, гарантирующее (при условии правильного распределения по островам) принятие этого решения.

Входные данные (input.txt)

В первой строке входного файла число островов N ($1 \leq N \leq 100$). Во второй строке через пробел численность населения островов — N чисел. Все числа — целые положительные, не превышающие 10 000.

Выходные данные(output.txt)

В первой строке выходного файла — минимально необходимое число сторонников решения.

Пример

input.txt

```
3
4 4 8
```

output.txt

```
6
```