

Лабораторная работа №2

Вариант 1.

Компания SPAM занимается рассылкой рекламных газет и буклетов в разные города. По договоренности с почтой, каждая посылка должна содержать строго определенное количество (N , где $1 < N < 100$) экземпляров рекламных изданий в сумме. Но есть еще проблема и принцип компании:

1) Два буклета не могут соприкасаться, из-за некачественного полиграфического покрытия они склеиваются и портятся. Газеты выполняют еще и роль прокладки между ними.

2) Каждая посылка должна быть уникальная, то есть отличаться от других порядком укладки или количеством различных типов изданий (буклет/газета). Это дополнительный рекламный ход компании, который состоит в том, что каждый клиент должен получать индивидуальную укладку или содержание посылки.

Необходимо заранее определить максимальное количество вариантов правильной комплектации посылки в зависимости от N .

Входные данные:

В начале файла "SPAM.IN" записано количество экземпляров изданий (некоторое N).

Выходные данные:

В первой строке файла "SPAM.OUT" нужно указать число возможных вариантов комплектации.

Пример 1.

Файл "SPAM.IN"4

Файл "SPAM.OUT"

8

Пример 2.

Файл "SPAM.IN"13

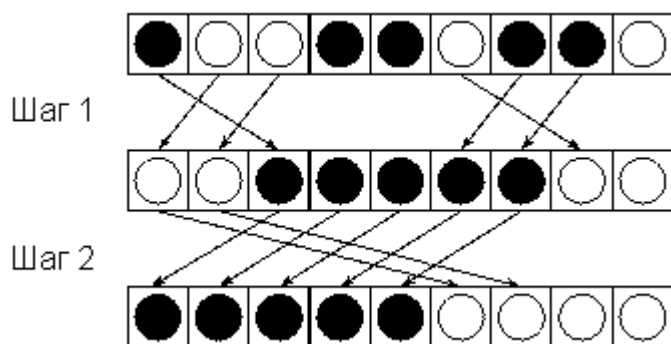
Файл "SPAM.OUT"

610

Вариант 2.

Последовательность клеток занумерована числами от 1 до N . В каждой клетке стоит либо черная, либо белая фишка. Группой назовем набор подряд стоящих фишек одного цвета, ограниченный с обеих сторон фишками другого цвета или концами последовательности. Следует переместить фишки так, чтобы они образовали не более двух групп.

Перемещение фишек описывается с помощью плана обмена, в котором используются понятия операция обмена и шаг. Операция обмена меняет местами две соседние группы фишек. Шаг состоит не более чем из K одновременно выполняемых обменов. Обмены можно совершать одновременно только тогда, когда в них участвуют разные группы. После каждого шага группы одного цвета, оказавшиеся рядом, объединяются. План обменов содержит описания шагов, выполняемых последовательно.



Напишите программу, определяющую план обменов, с помощью которого за наименьшее число шагов получается последовательность, состоящая не более чем из двух групп.

Формат входных данных

В первой строке входного файла записаны числа N и K ($1 \leq N \leq 100000$ и $1 \leq K \leq 10000$). Исходная расстановка фишек задается в последующих строках, содержащих N чисел (0 или 1), разделенных пробелами или переводами строк. При этом 0 соответствует черной фишке, 1 - белой.

Формат выходных данных

Выходной файл должен содержать описание шагов плана, по одному шагу на строке. Описание шага начинается с числа L - количества обменов на этом шаге. Затем для каждого обмена указывается минимальный номер клетки, в которой стоит фишка, участвующая в этом обмене. Последняя строка плана должна содержать одно число 0.

Примеры

fishes.in

9 3

1 0 0 1 1 0 1 1 0

fishes.out

2 1 6

1 1

0

fishes.in

3 1

1 1 0

fishes.out

0

Примечание

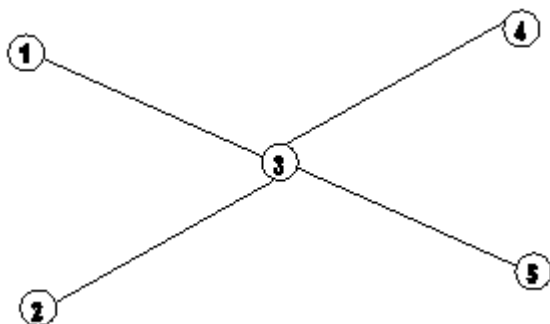
Требуется найти план, содержащий наименьшее число шагов, при этом общее число обменов может быть не минимальным.

Вариант 3.

В некотором городе есть метро, состоящее из N ($1 \leq N \leq 1000$) станций и M линий, соединяющих их. Каждая линия обеспечивает проезд между какими-то двумя станциями в обе стороны. Между любой парой станций проведено не более одной линии. Сеть метро построена таким образом, чтобы с каждой станции можно было проехать на каждую (возможно, через промежуточные станции). Назовем это свойство связностью метро.

В связи с изобретением принципиально нового вида транспорта метро стало убыточным, и его работу решили прекратить. На заседании мэрии города было постановлено закрывать каждый год по одной станции, но так, чтобы связность метро каждый раз сохранялась. При закрытии какой-либо станции, линии, ведущие из этой станции в другие, естественно, тоже перестают функционировать.

Задание. По введенной информации о сети метро разработать какой-либо порядок закрытия станций, при котором метро всегда будет оставаться связным. Например, пусть метро выглядит так, как показано на рисунке. Тогда станции можно закрывать, например, в порядке 1, 2, 4, 3, 5. А порядок 3, 1, 2, 4, 5 - не подходит, так как после закрытия 3-й станции метро распадется на четыре не связанных между собой части.



Ввод. Первая строка входного файла будет содержать числа N и M. В следующих M строках находится информация о линиях. Каждая из этих строк содержит через пробел числа A_i и B_i ($A_i B_i$) - две станции, которые соединяет i-я линия.

Вывод. Выходной файл должен состоять из N строк. Каждая строка должна содержать одно число - номер станции. Вывести станции нужно в порядке их закрытия.

Пример

input.txt

5 4

3 1

3 2

3 4

3 5

output.txt

1

2

4

3

5

Вариант 4.

С целью подготовки к проведению олимпиады по информатике мэр решил обеспечить надежным электроснабжением все школы города. Для этого необходимо провести линию электропередач от альтернативного источника электроэнергии "Майбуття" к одной из школ города (к какой неважно), а также соединить линиями электропередач некоторые школы между собой.

Считается, что школа имеет надежное электроснабжение, если она напрямую связана с источником "Майбуття", либо с одной из тех школ, которые имеют надежное электроснабжение.

Известна стоимость соединения между некоторыми парами школ. Мэр города решил выбрать одну из двух наиболее экономичных схем электроснабжения (стоимость схемы равняется сумме стоимостей соединений пар школ).

Задание

Напишите программу SCHOOLS, которая вычисляет стоимость двух наиболее экономных схем альтернативного электроснабжения школ.

Входные данные

В первой строке входного файла SCHOOLS.DAT находятся два натуральных числа, разделенных пробелом: N ($3 \leq N \leq 100$), количество школ в городе, и M - количество возможных соединений между ними. В каждой из последующих M строк находятся по три числа: A_i , B_i , C_i , разделенных пробелами, где C_i - стоимость прокладки линии электроснабжения ($1 \leq C_i \leq 300$) от школы A_i до школы B_i ($i = 1, 2, \dots, N$).

Пример входного файла

```
5 8
1 3 75
3 4 51
2 4 19
3 2 95
2 5 42
5 4 31
1 2 9
3 5 66
```

Выходные данные

В единственной строке выходного файла SCHOOLS.SOL должны содержаться два натуральных числа S_1 и S_2 , разделенных пробелом - две наименьшие стоимости схем ($S_1 \leq S_2$). $S_1 = S_2$ тогда и только тогда, когда существует несколько схем надежного электроснабжения наименьшей стоимости.

Пример выходного файла

```
110 121
```

Вариант 5.

Имеются гири с массами: 1 г, 2 г, ..., N г ($N \leq 500000$). Написать программу, распределяющую эти гири на максимально возможное количество пар так, чтобы суммарный вес гирь в каждой паре выражался простым числом.

Входные данные:

Входной файл input.txt содержит число N .

Выходные данные

В выходном файле output.txt выводится список найденных пар. Все числа в выходном файле разделяются пробелами и (или) символами перевода строки.

input.txt

```
7
```

output.txt

```
1 6
```

```
7 4
```

```
5 2
```

Вариант 6.

Странные времена настали в Лошине Янтарной Росы. Все куда-то бегут, что-то покупают-продают, постоянно норовя обмануть друг друга. Нет былого спокойствия. Смутное время не обошло и Монастырь Светлой Луны: Никогда еще не было такого, чтобы обычный торговец пытался обмануть монахов, боязнь гнева Будды останавливала его. Но и этот страх померк перед страстью наживы.

Мудрый Настоятель подозревает, что один из поставщиков Монастыря нечист на руку. Известно, что при подсчете стоимости товара он использует Калькулятор. Этот Калькулятор умеет не так уж и много... Все что он умеет это:

1. ввести число 1
2. удвоить текущее число
3. поменять в текущем числе первую и последнюю цифры.

Калькулятор умеет работать лишь с целыми числами от 1 до 10000.

Обычно Торговец привозит в Монастырь товар, затем, пользуясь Калькулятором, подсчитывает стоимость товара, называет сумму Настоятелю, и Настоятель оплачивает товар. Настоятель хочет узнать, не обманывает ли его Торговец, называя сумму, которая не может быть получена с помощью Калькулятора. Помогите ему в этом.

Ввод. В файле находится единственное число k - сумма, названная Торговцем ($1 \leq k \leq 10000$)

Вывод. Выведите "YES", если сумма может быть получена с помощью Калькулятора, и "NO" в противном случае.

Пример input.txt #1

8042

Пример output.txt #1

YES

Пример input.txt #2

3

Пример output.txt #2

NO

Вариант 7.

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

Формат входных данных

В первой строке содержится N ($3 \leq N \leq 1000$) - число вершин многоугольника. В последующих N строках идут координаты (X_i , Y_i) вершин многоугольника в порядке обхода по часовой стрелке. X_i и Y_i - целые числа, по модулю не превосходящие 1000000.

Формат выходных данных

В выходной файл вывести одно число - искомое число точек.

Примеры

e.in

4

-1 -1

-1 1

1 1

1 -1

e.out

1

e.in

3

0 0

0 2

2 0

e.out

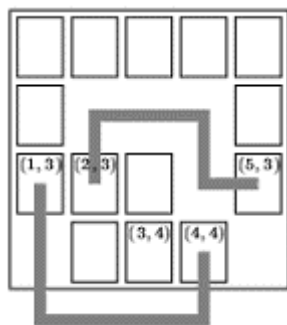
0

Вариант 8.

Вы являетесь одним из разработчиков новой компьютерной игры. Игра происходит на прямоугольной доске, состоящей из $W \times N$ клеток. Каждая клетка может либо содержать, либо не содержать фишку (см. рисунок).

Важной частью игры является проверка того, соединены ли две фишки путем, удовлетворяющим следующим свойствам:

1. Путь должен состоять из отрезков вертикальных и горизонтальных прямых.
2. Путь не должен пересекать других фишек.
3. При этом часть пути может оказаться вне доски. Например:



Фишки с координатами (1,3) и (4,4) могут быть соединены. Фишки с координатами (2,3) и (5,3) тоже могут быть соединены. А вот фишки с координатами (2,3) и (3,4) соединить нельзя - любой соединяющий их путь пересекает другие фишки.

Вам необходимо написать программу, проверяющую, можно ли соединить две фишки путем, обладающим вышеуказанными свойствами, и, в случае положительного ответа, определяющую минимальную длину такого пути (считается, что путь имеет изломы, начало и конец только в центрах клеток (или "мнимых клеток", расположенных вне доски), а отрезок, соединяющий центры двух соседних клеток, имеет длину 1).

Формат входных данных

Первая строка входного файла содержит два натуральных числа: W - ширина доски, N - высота доски ($1 \leq W, N \leq 75$). Следующие N строк содержат описание доски: каждая строка состоит ровно из W символов: символ "X" (заглавная английская буква "икс") обозначает фишку, символ "." (точка) обозначает пустое место. Все остальные строки содержат описания запросов: каждый запрос состоит из четырёх натуральных чисел, разделённых пробелами - X_1, Y_1, X_2, Y_2 , причём $1 \leq X_1, X_2 \leq W, 1 \leq Y_1, Y_2 \leq N$. Здесь (X_1, Y_1) и (X_2, Y_2) - координаты фишек, которые требуется соединить (левая верхняя клетка имеет координаты (1,1)). Гарантируется, что эти координаты не будут совпадать (кроме последнего запроса; см. далее). Последняя строка содержит запрос, состоящий из четырёх чисел 0; этот запрос обрабатывать не надо. Количество запросов не превосходит 20.

Формат выходных данных

Для каждого запроса необходимо вывести одно целое число на отдельной строке - длину кратчайшего пути, или 0, если такого пути не существует.

Примеры

```
g.in
5 4
XXXXX
X...X
XXX.X
.XXX.
2 3 5 3
```

```

1 3 4 4
2 3 3 4
0 0 0 0
g.out
5
6
0
g.in
4 4
XXXX
.X..
..X.
X...
1 1 3 1
0 0 0 0
g.out
4

```

Вариант 9.

Дано N квадратов на плоскости ($1 < N < 100$), причем стороны квадратов параллельны осям координат. Определить площадь области, которую определяют эти квадраты в совокупности (точка принадлежит этой области, если она находится хотя бы в одном из квадратов).

Формат входных данных

В первой строке входного файла находится число N .

В последующих N строках находятся тройки чисел (x, y, a) , каждая из которых описывает один квадрат следующим образом: (x, y) - координаты центра квадрата, a - длина его стороны. Все числа - целые. $-30000 < x, y < 30000$. $0 < a < 1000$.

Формат

выходных

данных

В выходной файл следует вывести площадь области, покрытой данными квадратами.

Округлить значение до сотых и вывести ровно два знака после десятичной точки.

Пример ввода

```

3
0 0 2
1 1 2
-3 1 5

```

Правильный вывод для примера

```

32.00

```

Вариант 10.

В одном государстве имеется N городов. Некоторые города соединены дорогами, причем для любых двух городов A и B выполняется следующее свойство: существует ровно один способ попасть из города A в город B если можно перемещаться только по дорогам и не разрешается проезжать по одной и той же дороге более одного раза.

Недавно президента этой страны заинтересовал вопрос: какие три города являются наиболее удаленными друг от друга. А именно, назовем взаимной удаленностью друг от друга трех городов A , B и C минимальное количество дорог, которое необходимо

использовать, чтобы доехать от А до В, затем от В до С и затем от С до А (при этом разрешается использовать одну и ту же дорогу в различных путешествиях).

Требуется найти три города, для которых взаимная удаленность друг от друга будет максимальной.

Например, для пяти городов, соединенных дорогами так, как это показано на рисунке 1, три наиболее удаленных друг от друга города - это города 1, 2 и 5 (взаимная удаленность равна $2 + 3 + 3 = 8$), а для городов на рисунке 2 - это любые три города, выбранные из множества $\{1, 2, 4, 5\}$ (удаленность $2 + 2 + 2 = 6$).

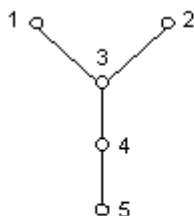


рисунок 1

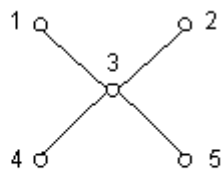


рисунок 2

Формат входных данных

Первая строка входного файла содержит число N - количество городов ($3 \leq N \leq 1000$). Следующие N строк содержат описания городов. Описание i -го города сначала содержит K_i - количество городов, с которыми он соединен дорогами ($1 \leq K_i < N$), а затем K_i чисел - номера городов, с которыми он соединен.

Гарантируется, что входные данные корректны - то есть если есть дорога из города А в город В, то есть и дорога из города В в город А, причем для всех пар городов выполняется свойство, указанное в условии задачи.

Формат выходных данных

Выведите в выходной файл три различных числа - номера трех наиболее удаленных друг от друга городов в произвольном порядке. Если решений несколько, выведите любое из них.

Примеры

three.in

5

1 3

1 3

3 1 2 4

2 3 5

1 4

three.out

1 2 5

three.in

5

1 3

1 3

4 1 2 4 5

1 3

1 3

three.out

1 2 4

Вариант 11.

Государство Флатландия представляет собой прямоугольник размером $M * N$, состоящий из единичных квадратиков. Флатландия разделена на K провинций ($2 \leq K \leq 100$). Каждая провинция представляет собой связное множество квадратиков, т.е. из каждой точки провинции можно дойти до любой другой ее точки, при этом разрешается переходить с квадратика на квадратик, только если они имеют общую сторону (общей вершины недостаточно). Во Флатландии нет точки, которая граничила бы более чем с тремя провинциями (т.е. четыре квадратика, имеющие общую вершину, не могут принадлежать четырем разным провинциям).

Каждая провинция имеет свой символ. Столица Флатландии находится в провинции, которой принадлежит символ A (заглавная латинская буква A). Провинция называется пограничной, если она содержит граничные квадратики. Провинция, в которой находится столица Флатландии, не является пограничной.

Король соседнего с Флатландией королевства Ректиландия решил завоевать Флатландию. Для этого он хочет захватить столицу Флатландии. Однако он знает, что сил его армии недостаточно, чтобы сделать это сразу. Поэтому сначала он хочет окружить столичную провинцию, чтобы ослабить силы противника долгой блокадой, а потом захватить столицу.

Чтобы окружить провинцию, требуется захватить все провинции, с которыми она граничит. Две провинции граничат, если существует два квадратика, имеющие общую сторону, один из которых принадлежит первой из них, а другой - второй. Чтобы захватить провинцию, надо чтобы выполнялось одно из двух условий: либо она пограничная, либо граничит с какой-либо уже захваченной провинцией.

Чтобы сберечь силы своей армии, король Ректиландии хочет установить блокаду столичной провинции, захватив как можно меньше провинций. Помогите ему выяснить, сколько провинций требуется захватить. Захватывать столичную провинцию нельзя, поскольку для этого сил армии Ректиландии пока недостаточно.

Формат входных данных

Первая строка содержит M и N ($3 \leq M, N \leq 200$). Следующие M строк содержат N символов каждая и задают карту Флатландии. Символ, находящийся в $i+1$ -й строке входного файла на j -м месте, представляет собой символ провинции, которой принадлежит квадратик (i, j) . Все символы имеют ASCII-код больше 32 (пробела).

Формат выходных данных

Выведите в выходной файл единственное число - количество провинций, которые требуется захватить. Если установить блокаду невозможно, выведите "-1".

Примеры

input.txt

5 6

BBBBBZ

BCCCBZ

BCAAbbZ

BDDDbZ

33333Z

output.txt

4

Вариант 12.

Надеемся, что вам знакома следующая игра. Выбирается слово и из его букв составляются другие осмысленные слова, при этом каждая из букв исходного слова может быть использована не более такого количества раз, которое она в нем встречается.

Напишите программу, помогающую играть в эту игру.

Входные данные

В первой строке входного файла записано выбранное для игры слово. В последующих строках задан "словарь" - множество слов, которые мы считаем осмысленными. Их количество не превышает 1000. Слово - это последовательность не более чем 255 маленьких латинских букв. Каждое слово записано в отдельной строке. Заканчивается словарь словом "ZZZZZZ" (состоящим из заглавных букв), которое мы не будем считать осмысленным. Слова в словаре, как это ни странно, не обязательно располагаются в алфавитном порядке.

Выходные данные

В выходной файл необходимо выдать список слов, которые можно получить из исходного слова. Слова должны быть выданы в том же порядке, в котором они встречаются в словаре. Каждое слово должно быть записано в отдельной строке. Список слов должен заканчиваться все тем же неосмысленным словом "ZZZZZZ".

Примеры входного и выходного файлов

C.IN

Soundblaster

Sound

Sound

Blaster

Soundblaster

Master

Last

Task

Sos

Test

Bonus

Done

ZZZZZZ

C.OUT

Sound

Sound

Blaster

Soundblaster

Last

Sos

Bonus

Done

ZZZZZZ

C.IN

Windowsmustdie

Summer

Informatics

School

Rules

Olympiadisstarting

Goodbymylovegoodby

Exit

ZZZZZZ

C.OUT

ZZZZZZ

Вариант 13.

Пользователь сети Интернет подписан на несколько разных списков рассылки, которые высылают ему по электронной почте сообщения на определенные темы. Для удобства пользователь создал себе набор папок, каждая из которых соответствует одной из тем. Перед тем, как читать сообщения он копирует их в соответствующую папку.

Почтовая программа, установленная на компьютере пользователя, позволяет за одну "операцию" переносить из "списка новых сообщений" в соответствующую папку:

- Одно сообщение из любого места списка
- Несколько сообщений, идущих в списке подряд, и относящихся к одной теме

Переносить можно не обязательно начиная с начала "списка новых сообщений". Пользователю необходимо перенести все новые сообщения в соответствующие им папки за наименьшее количество операций.

Пример. Пусть пользователь подписан на рассылки анекдотов, веселых историй, спортивных новостей и прогноза погоды. Пусть "список новых сообщений" при некотором вхождении в почтовую программу имел следующий вид: (Анекдоты, Спортивные новости, Прогноз погоды, Спортивные новости, Веселые истории, Веселые истории, Спортивные новости)

Список папок Список новых сообщений

1 Анекдоты	1 Анекдоты
2 Веселые истории	3 Спортивные новости
3 Спортивные новости	4 Прогноз погоды
4 Прогноз погоды	3 Спортивные новости
2 Веселые истории	
2 Веселые истории	
3 Спортивные новости	

Переносить сообщения в папки он может следующим образом: сначала два сообщения на тему "Веселые истории". Тогда он получит следующий список: (Анекдоты, Спортивные новости, Прогноз погоды, Спортивные новости, Спортивные новости). Потом перенести сообщения о прогнозе погоды, после этого "Анекдоты", а потом, одновременно, все три сообщения о спортивных новостях. На это он потратит 4 "операции".

Задание: Напишите программу EMAIL которая будет вычислять минимальное количество "операций", с помощью которых можно перенести все новые сообщения в папки. Для удобства каждой теме присваивается номер.

Входные данные: Единственная строка входного файла EMAIL.DAT содержит число N ($0 < N < 200$), отвечающее количеству новых сообщений и N целых чисел, которые соответствуют сообщениям, и являются номерами тем, которым эти сообщения принадлежат.

Пример входных данных

7 1 3 4 3 2 2 3

Выходные данные В первой строке выходного файла EMAIL.SOL должно находиться минимальное число операций для данных, приведенных во входном файле.

Пример выходных данных

4

Вариант 14.

Трехмерная фигура состоит из единичных кубиков. По фигуре можно построить ее фронтальную и правую проекции. Очевидно, что по этим двум проекциям не всегда можно восстановить фигуру.

Задача

Напишите программу CUBES, которая получает на вход фронтальную и правую проекции фигуры и определяет минимальное и максимальное количество кубиков, которое можно было бы использовать для построения фигуры с заданными проекциями.

Входные данные

В первой строке входного файла CUBES.DAT находятся три числа N , M и K , которые задают размеры проекций ($1 < N, M, K < 100$). Далее задаются две проекции: сначала фронтальная, а затем правая. Проекция задается N строками, каждая из которых состоит из чисел 0 и 1, разделенных пробелами. Для фронтальной проекции таких чисел будет M , а для правой - K . 0 означает свободную клетку проекции, 1 - заполненную.

Пример входных данных

```
2 2 3
1 0
1 1
0 0 1
1 1 1
```

Выходные данные

В единственной строке выходного файла CUBES.SOL должно находиться два числа: минимальное и максимальное число кубиков, которые можно было бы использовать для построения фигуры с заданными проекциями.

Пример выходных данных

```
4 7
```

Вариант 15.

В МЧС поступило сообщение о возможном лесном пожаре в заданном квадрате тайги. Для поиска места возгорания было послано N самолетов. Однако ни один из экипажей пожара не обнаружил.

Известно, что с самолета видна полоса тайги, границы которой находятся на расстоянии 50 км справа и слева от той линии на поверхности Земли, над которой пролетает самолет (см. рисунок), причем точки, находящиеся на расстоянии ровно 50 км от этой линии, все еще видны. Донесение с каждого самолета содержало информацию о том, в каких двух различных точках (x_b, y_b) и (x_e, y_e) самолет входил в заданный квадрат и покидал его соответственно. Между этими точками самолет двигался строго по прямой.

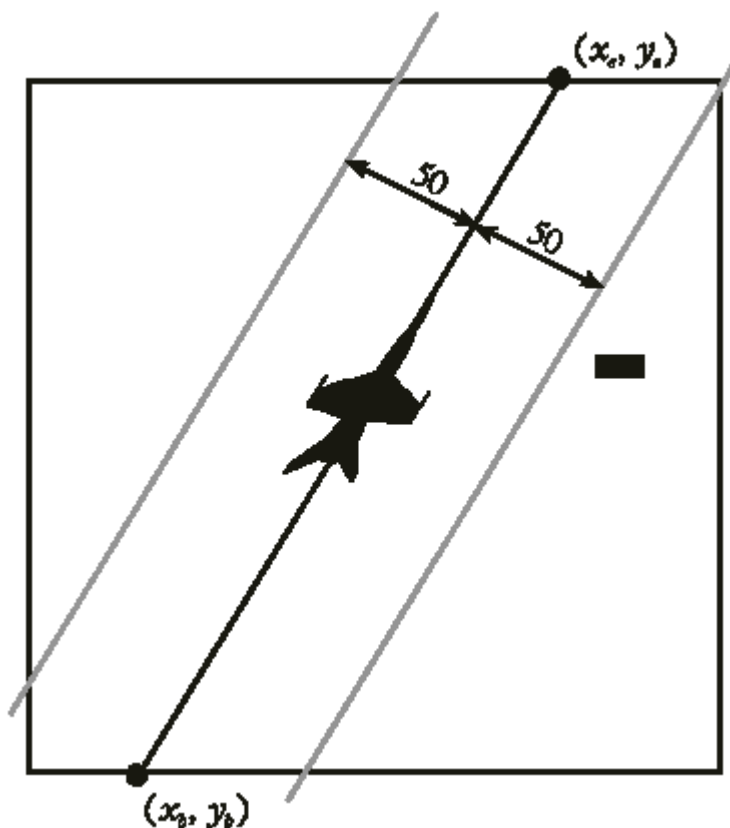


Рис. 5

Требуется

Написать программу, которая определит, действительно ли весь заданный квадрат тайги был просмотрен с самолетов. Если это не так, то программа должна находить координаты какой-нибудь точки, лежащей внутри или на границе квадрата и не попавшей ни в одну из просмотренных полос.

Входные данные

Входной файл с именем FIRE.IN состоит из $N + 2$ строк.

В первой строке записано натуральное число L - размер заданного квадрата тайги в километрах ($0 < L \leq 1000$). Во второй строке - натуральное число N ($1 \leq N \leq 100$) - количество самолетов. В каждой из последующих N строк записано донесение с самолета - четыре вещественных координаты x_b , y_b , x_e , y_e . Координаты заданы в километрах. Стороны квадрата тайги параллельны осям координат, его левый нижний угол находится в точке с координатами $(0, 0)$, а правый верхний - в точке (L, L) .

Выходные данные

Выходной файл с именем FIRE.OUT должен содержать одну строку. Если заданный квадрат был просмотрен полностью, то эта строка должна состоять из слова ОК, написанного заглавными латинскими буквами. В противном случае в этой строке должны быть записаны через пробел координаты x и y какой-либо точки, которая не попала ни в одну из просмотренных полос. Координаты нужно выводить в километрах с ошибкой не более одного метра.

Пример входного файла

245

1

26.1 0 193.568 245

Пример выходного файла для приведенного примера входного файла

155.123 100

Примечание. Будут отдельно оцениваться решения, сделанные в предположении, что каждый самолет летел параллельно одной из сторон квадрата.