1. Шифр

1 секунда 30 очков

Рассмотрим простую систему шифрования текста. Для шифровки выбирается некое ключевое слово (ключ), и используется нижеуказанная таблица из 26×26 букв, где в первой строке латинский альфавит и каждая следующая строка получается сдвигом предыдущей на одно место налево:

	a	**b**	**c**	...	**z**
a	a	b	c	...	z
b	b	c	d	...	a
⋮	⋮	⋮	⋮		⋮
z	z	a	b	...	y

Чтобы зашифровать данный исходный текст с помощью ключа, i-тая буква текста заменяется на букву, которая находится в таблице в столбце, отмеченном i-той буквой текста и в ряду, отмеченном i-той буквой ключа. Все символы в тексте, которые не являются буквами, остаются неизменными. В случае, если длина текста больше длины ключа, ключ повторяется циклически, т.е. после последней буквы ключа используется опять первая, затем вторая, и т.д.

Требуется разработать алгоритм расшифровки вышеописанным образом зашифрованного текста, а также написать программу осуществляющую шифровку и расшифровку.

Входные данные. На первой строке текстового файла `kood.sis` задано имя операции, которую необходимо осушествить: слово `KODEERIDA` означает, что текст следует зашифровать, слово `DEKODEERIDA` указывает, что текст следует расшифровать. На второй строке дано ключевое слово состоящее из 1 до 250 прописных латинских букв. На третьей строке файла дано число строк N ($1 \leqslant N \leqslant 250$) в исходном тексте, и на следующих N строках сам текст. Каждая строка текста не длиннее 250 символов и состоит из прописных латинских букв, знаков препинания и пробелов.

Выходные данные. В текстовый файл `kood.val` вывести в точности N строк: результат шифровки или расшифровки исходного текста.

Пример.

`kood.sis`
```
KODEERIDA
salasona
2
esimene rida
teine rida
```

`kood.val`
```
wstmwbr radl
twwae jioa
```

Пример.

`kood.sis`
```
DEKODEERIDA
salasona
2
wstmwbr radl
twwae jioa
```

`kood.val`
```
esimene rida
teine rida
```

2. Анализ четырехугольника

1 секунда 30 очков

Землемер сообщил хозяину участка земли координаты четырёх углов его участка. Хозяин желает узнать какова форма участка, так как от этого зависит его цена. Более дорогими являются участки правильной формы. Хозяину интересно, является ли его участок (в порядке предпочтения) квадратом, прямоугольником, ромбом, параллелограмом, трапецией или иным четырёхугольником.

Требуется написать программу, которая проверяет по координатам четырёх углов, предоставленных землемером, какого рода четырёхугольник получится если соединить отрезками прямых первую точку со второй, вторую с третьей, третью с четвёртой и четвёртую с первой.

Входные данные. На четырёх строках текстового файла `neli.sis` даны координаты углов. Каждая строка содержит два целых числа разделённых пробелом x_i y_i ($|x_i| \leqslant 10\,000$, $|y_i| \leqslant 10\,000$).

Выходные данные. На единственной строке текстового файла `neli.val` вывести одно слово, опи-

сывающее форму участка: `RUUT` (квадрат), `RISTKULIK` (прямоугольник), `ROMB` (ромб), `ROOPKULIK` (параллелограм), `TRAPETS` (трапеция), `MUU` (другой четырёхугольник), или `VIGA` (ошибка), если точки в указанном порядке не образуют самонепересекающегося четырёхугольника.

Пример.

`neli.sis`
```
1 1
1 3
3 3
3 1
```

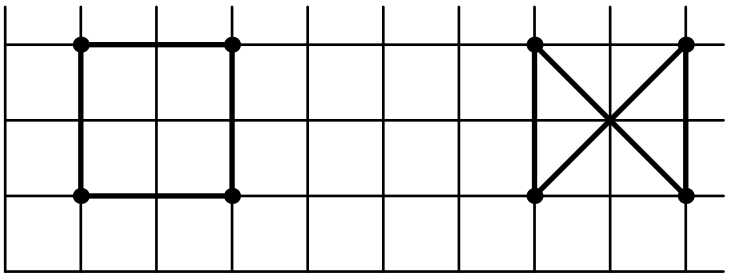
`neli.val`
```
RUUT
```

Пример.

`neli.sis`
```
7 1
7 3
9 1
9 3
```

`neli.val`
```
VIGA
```

Рисунок иллюстрирует приведённые в примерах “участки”.



3. Дорожная карта

2 секунды — 40 очков

В одной стране есть N городов, а между ними M дорог. Каждая дорога соединяет между собой два города и никакие две дороги не пересекаются (кроме как в самих городах). Требуется написать программу, которая для каждого города найдёт расстояние до самого близкого к нему и самого далёкого от него городов. Предполагается, что передвигаться из города в город можно только по дорогам и расстоянием между городами считается длина наикратчайшего пути.

Входные данные. На первой строке текстового файла `teed.sis` дано количество городов N ($2 \leqslant N \leqslant 1000$) и дорог M ($1 \leqslant M \leqslant 10\,000$). Города пронумерованы $1 \ldots N$. На каждой из следующих M строках дано по 3 целых числа: описание дороги в формате l_1 l_2 d ($1 \leqslant l_1, l_2 \leqslant N$, $l_1 \neq l_2$, $1 \leqslant d \leqslant 1000$), что означает что данная дорога соединяет города l_1 и l_2, причём длина дороги d километров. Известно что из каждого города можно добраться до каждого другого.

Выходные данные. В текстовый файл `teed.val` вывести ровно N строк, на каждой строке по два разделённых пробелом числа. На строке i должно быть выведено расстояние от города i до ближайшего к нему города d_i и расстояние до самого далёкого от него города D_i.

Пример.

`teed.sis`
```
4 4
1 3 10
1 4 15
2 3 10
3 4 12
```

`teed.val`
```
10 20
10 22
10 12
12 22
```