

ЮЖНО-УРАЛЬСКИЙ  
**СОТОВЫЙ**  
ТЕЛЕФОН

Студенческий командный чемпионат мира по программированию ACM 2003-2004  
Северо-восточный Европейский регион  
Восточный (Уральский) подрегион  
Первый тур

*Во всех задачах ввод данных производится из файла INPUT.TXT, а вывод результатов - в файл OUTPUT.TXT. Формат входного файла соответствует спецификации, дополнительные проверки не нужны. Все строки, в том числе последняя, оканчиваются символом перехода на новую строку.*

*Время теста указано для процессора Celeron 600 МГц.*

*Персонажи взяты из книги С. Лема "Кибериада".*

1. Кодовый замок	5 сек
3. Меандр	5 сек
4. Н	5 сек
6. Замощение	5 сек
7. Бочка	5 сек
8. Лифты	5 сек

## 1. Кодовый замок

<i>Входной файл</i>	<i>INPUT.TXT</i>
<i>Выходной файл</i>	<i>OUTPUT.TXT</i>
<i>Ограничение времени</i>	<i>5 сек</i>

Однажды конструктор Трурль заглянул в гости к Клапауцию, и тот похвастался новым кодовым замком, стоящим на двери секретной лаборатории.

– Вот, смотри – на замке несколько кнопок с буквами. Можно нажимать кнопки сколько угодно – дверь не откроется, пока  $K$  (я установил  $K$  равным трем) последних нажатых букв не совпадут с правильной комбинацией. Буквы в этой комбинации не должны повторяться, так как в замок встроен специальный защитный механизм. Если среди любых  $K$  последних введенных букв есть хотя бы две одинаковые, то на мой пейджер поступает сообщение, что кто-то пытается открыть дверь, подбирая код.

Через несколько дней, когда Клапауций отправился на собрание Артели Свободных Механиков, Трурль проник в его дом, чтобы ознакомиться с новыми изобретениями Клапауция в его секретной лаборатории. Чтобы открыть кодовый замок нужно было всего лишь перебрать все комбинации, не вызвав тревоги. И следовало бы поторопиться, так как Клапауций мог явиться с собрания в любой момент.

Напишите программу, которая по количеству букв в комбинации для открытия кодового замка и количеству кнопок на замке определит минимальную последовательность нажатий кнопок замка, которая содержит все возможные комбинации и не вызывает тревоги.

### *Ввод*

Во входном файле в первой строке содержатся два целых числа  $K$  и  $N$  ( $1 \leq K \leq N \leq 26$ ,  $N^K \leq 60000$ ), разделенных пробелом – количество букв в комбинации для открытия кодового замка и количество кнопок на замке. Кнопки обозначаются последовательными прописными латинскими буквами, начиная с А.

### *Вывод*

В выходной файл вывести минимальную последовательность нажатий кнопок замка, содержащую все возможные комбинации. Если существует несколько вариантов, то вывести один (любой) из них. Если перебрать все комбинации без тревоги не удастся, то вывести сообщение IMPOSSIBLE.

*Пример INPUT.TXT:*

3 4

*OUTPUT.TXT для примера:*

ABCABDACBACDBADCBCADBCDAB

### 3. Меандр

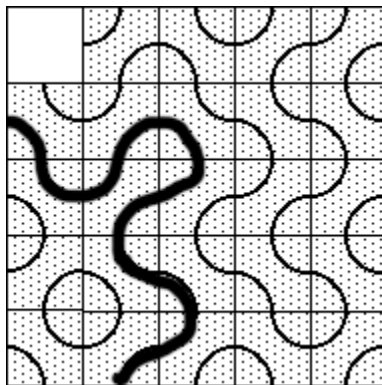
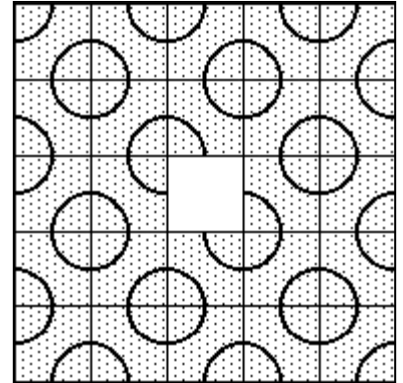
Входной файл	INPUT.TXT
Выходной файл	OUTPUT.TXT
Ограничение времени	5 сек

После завтрака друзья решили сыграть в свою любимую игру “меандр”, требующую от игроков недюжинного интеллекта и необычайного хитроумия.

Трурль сделал очередной ход, но через секунду воскликнул:

- Черт, не надо было мне так ходить.
- Трурль потянулся к доске.
- Нет уж, перехаживать нельзя, – возразил Клапауций.
- Теперь ты можешь выиграть, если только найдешь как.

Трурль откинулся в кресле и начал фальшиво насвистывать. Клапауций пытался сообразить, какой из восьми возможных ходов позволит ему добиться победы. “Хорошо бы воспользоваться помощью компьютера, он бы быстро рассмотрел все варианты и подсказал верный ход”. И Клапауций представил, что компьютер в форме кольца надет у него на палец, и стоит только протянуть руку к нужной фишке, как компьютер...



– Хватит руками над доской водить, ходи, наконец! – Голос Трурля прервал мечты Клапауция.

В игре “меандр” 24 одинаковые плитки раскладывают на доске с бортиками так, как показано на рисунке справа вверху. Одна клетка остается свободной. Игроки делают ходы поочередно, перемещая одну плитку или столбик из двух, трех или четырех плиток, расположенных по горизонтали или вертикали. Плитки перемещаются по поверхности доски, их нельзя поворачивать. Игра продолжается до тех пор, пока один из игроков не выиграет, сложив узор, в котором по крайней мере три плитки образуют непрерывную линию, соединяющую два края доски (либо противоположные, либо смежные, сходящиеся в одном углу). Такой узор показан на рисунке слева (победная линия выделена жирной линией).

Напишите программу, которая подсказывает игроку, что уже получен победный узор, или каким ходом можно получить победный узор.

#### Ввод

Во входном файле содержится пять строк по пять символов в каждой строке. Символом ‘\*’ обозначается



свободная клетка. Символом ‘/’ обозначается плитка , а символом ‘\’ – плитка . Во входном файле ровно один символ ‘\*’ и по 12 символов ‘/’ и ‘\’.

#### Вывод

В выходной файл вывести или сообщение WIN, если узор уже является победным, или ход, который создает из заданного непобедного узора победный узор, в форме направление движения (L – влево, R – вправо, D – вниз, U – вверх) и количество сдвигаемых плиток (от 1 до 4), например, U2 (если существует несколько вариантов, то вывести один (любой) из них), или сообщение NO WIN, если задан непобедный узор и добиться победы одним ходом невозможно.

Пример INPUT.TXT:

```
*\\//
\\\\
\\///
/\\\\
\\//\
```

OUTPUT.TXT для примера:

```
WIN
```

**4. Н**

<i>Входной файл</i>	<i>INPUT.TXT</i>
<i>Выходной файл</i>	<i>OUTPUT.TXT</i>
<i>Ограничение времени</i>	<i>5 сек</i>

Чтобы помочь Трурлю проверить машину, напишите программу, которая в заданном тексте выделяет слова на букву «н». Программа должна сделать прописной первую буквы всех слов, начинающихся с буквы «н» и длиной не менее трех букв. Под словом будем понимать последовательность русских букв, перед и после которой находятся символы, отличные от русских букв.

***Ввод***

Во входном файле содержится текст – не менее одной строки длиной не более 255 символов. Текст состоит только из строчных и прописных русских букв (используется кодировка cp866 (DOS)) и знаков препинания.

***Вывод***

В выходной файл вывести текст из входного файла, превращая первые буквы слов, начинающихся с “н”, в прописные. Разбиение текста на строки, пробелы и знаки препинания должны сохраниться.

***Пример INPUT.TXT:***

Конструктор Трурль создал однажды машину, которая умела делать всё на букву “Н”. Закончив эту машину, он для пробы заставил её сделать нитки, потом связать из ниток носки, одеть их на ноги, а затем бросить всё это в нору, окружённую незабудками и наличниками. Машина выполнила задание безукоризненно, но Трурль ещё не был уверен в её исправности и стал думать, какие вещи на букву “Н” можно приказать сделать машине.

***OUTPUT.TXT для примера:***

Конструктор Трурль создал однажды машину, которая умела делать всё на букву “Н”. Закончив эту машину, он для пробы заставил её сделать Нитки, потом связать из Ниток Носки, одеть их на Ноги, а затем бросить всё это в Нору, окружённую Незабудками и Наличниками. Машина выполнила задание безукоризненно, но Трурль ещё не был уверен в её исправности и стал думать, какие вещи на букву “Н” можно приказать сделать машине.

**6. Замощение**

<i>Входной файл</i>	<i>INPUT.TXT</i>
<i>Выходной файл</i>	<i>OUTPUT.TXT</i>
<i>Ограничение времени</i>	<i>5 сек</i>

Конструктор Клапауций решил замостить свой прямоугольный дворик плиткой. Для замощения Клапауций приготовил квадратные плитки двух разных размеров. При мощении плитки не должны накладываться друг на друга, выходить за пределы дворика, и не должно остаться незамощенных мест. Кроме того, Клапауций хотел бы как можно меньше использовать плитки первого типа, так как ему не нравится их цвет.

Напишите программу, которая определяет, возможно ли замощение двора плитками двух заданных размеров, и, если возможно, то какое количество плиток каждого размера следует взять.

**Ввод**

Во входном файле в первой строке содержится четыре целых числа, разделенных пробелами: размеры дворика  $W$  и  $H$  в сантиметрах ( $1 \leq W \leq 10^5$ ,  $1 \leq H \leq 10^5$ ) и размеры плиток  $R_1$  и  $R_2$  в сантиметрах ( $1 \leq R_1 \leq 1000$ ,  $1 \leq R_2 \leq 1000$ ).

**Вывод**

В выходной файл в первой строке вывести два целых числа, разделенных пробелом – количество плиток размером  $R_1$  и количество плиток размером  $R_2$ , которые потребуются для замощения дворика. Количество плиток первого типа должно быть минимальным. Если замощение невозможно, то нужно вывести сообщение IMPOSSIBLE.

*Пример INPUT.TXT:*

7 4 1 2

*OUTPUT.TXT для примера:*

4 6

## 7. Бочка

Входной файл	INPUT.TXT
Выходной файл	OUTPUT.TXT
Ограничение времени	5 сек

Как-то раз Трурль заглянул к Клапауцию и увидел, что тот бежит с огромной линейкой вокруг большой бочки, лежащей на боку.

– Что ты делаешь? – удивился Трурль.

– Да вот, хочу нанести разметку, чтобы всегда знать, сколько литров машинного масла у меня в запасе.

На днище бочки была прорезана вертикальная щель, закрытая стеклом. На нем Клапауций собирался делать отметки.

– Чтобы нанести отметку, например, 5000 литров, мне нужно вычислить на каком уровне будет находиться поверхность масла.

– Ну, это просто, делим объем на длину бочки, получаем площадь. Потом... – тут Трурль замолчал и начал лихорадочно чертить формулы в блокноте. Исписав несколько листов, Трурль сказал, что тут необходим принципиально другой подход.

Сказано – сделано! Трурль и Клапауций собрали под электронным микроскопом из отдельных атомов несколько миллионов созданных ненамного крупнее микробов, нарекли их ангстремиками и привили им склонность к командным соревнованиям по программированию. Поместив культуру ангстремиков в цивилизационный инкубатор, конструкторы отправились обедать. Так как время в инкубаторе сжималось в 480 тысяч раз, через полчаса знания и умения ангстремиков стали достаточными для решения задачи о бочке.

Напишите программу, которая рассчитает расположение отметок для нескольких заданных объемов масла в бочке. Предполагается, что бочка лежит абсолютно горизонтально.

### Ввод

Во входном файле в первой строке содержатся три целых числа, разделенных пробелами – диаметр бочки в метрах  $D$  ( $1 \leq D \leq 10$ ), длина бочки в метрах  $L$  ( $1 \leq L \leq 10$ ) и количество отметок  $N$  ( $1 \leq N \leq 10$ ). Далее следует  $N$  строк. В каждой строке содержится одно целое число – объем масла в бочке в литрах  $V_i$ . Объем масла  $V_i$  не превышает емкости бочки.

### Вывод

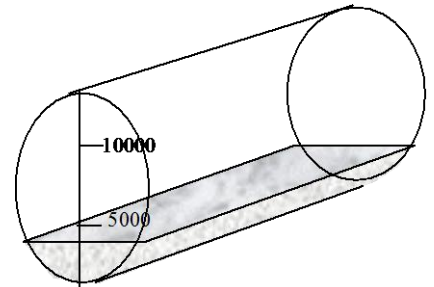
В выходной файл вывести  $N$  строк. В каждой строке выводится уровень в метрах, на котором будет находиться поверхность  $V_i$  литров масла, с 7 десятичными знаками.

Пример INPUT.TXT:

```
2 4 2
10000
5000
```

OUTPUT.TXT для примера:

```
1.4842576
0.8389023
```



## 8. Лифты

<i>Входной файл</i>	<i>INPUT.TXT</i>
<i>Выходной файл</i>	<i>OUTPUT.TXT</i>
<i>Ограничение времени</i>	<i>5 сек</i>

Чтобы получить разрешение на строительство Омнигенерического Ультимата в полом нутре Рапундры, огромной луны недотяпов, конструктор Клапауций отправился в столицу их планеты, где обнаружил, что все министерства планеты размещались в новом многоэтажном здании, и теперь для получения необходимых подписей и печатей посетителям не требовалось ездить в разные концы города.

Здание было снабжено сложной системой лифтов. Лифты не управлялись пассажирами, а двигались автоматически. Например, один из лифтов останавливался для высадки и посадки пассажиров через каждые 5 этажей, а другой – через каждые 12 этажей. При этом время поездки на любом из лифтов не зависело от количества этажей и равнялось 1 пякунде (единица измерения времени недотяпов, около 5 секунд). Например, пусть лифт в 10-этажном здании, который проезжает за поездку 3 этажа, находится на пятом этаже и движется вниз. Тогда через пякунду он будет на втором этаже, затем опять на пятом, потом на восьмом, затем спустится на пятый и так далее. Движение лифтов было синхронизировано таким образом, что в конце каждой пякунды все лифты останавливались, и можно было выйти и перейти в другой лифт, если он в этот момент был на том же этаже. Кроме того, в здании имела лестница, по которой можно было за одну пякунду подняться или спуститься на один этаж. Можно также стоять на этаже и ждать лифта.

Напишите программу, которая поможет Клапауцию быстро путешествовать по зданию.

### *Ввод*

Во входном файле в первой строке содержатся четыре целых числа, разделенных пробелами – количество этажей в здании  $H$  ( $1 < H \leq 1000$ ), количество лифтов  $N$  ( $1 \leq N \leq 100$ ), номер этажа  $A$  ( $1 \leq A \leq H$ ), на котором находится Клапауций, и номер этажа  $B$  ( $1 \leq B \leq H$ ), на который требуется попасть Клапауцию. Далее следует  $N$  строк, содержащих по три целых числа, разделенных пробелами – номер этажа  $P_i$  ( $1 \leq P_i \leq H$ ), на котором находится лифт в начальный момент путешествия Клапауция, число этажей  $S_i$  ( $1 \leq S_i \leq H/2$ ), на которое лифт поднимается или спускается за одну пякундную поездку, и направление движения лифта в следующий интервал времени  $D_i$  ( $D_i = 1$ , если лифт поедет вверх,  $D_i = -1$ , если лифт поедет вниз).

### *Вывод*

В выходной файл вывести одну строку, содержащую два целых числа, разделенных пробелом – время в пякундах, необходимое Клапауцию, чтобы добраться до нужного этажа, и количество этажей, которое Клапауцию придется пройти по лестнице. Требуется минимизировать в первую очередь время путешествия, а затем количество этажей, на которое Клапауцию придется подняться или спуститься без помощи лифта.

### *Пример INPUT.TXT:*

```
10 2 1 9
5 3 -1
1 1 1
```

### *OUTPUT.TXT для примера:*

```
4 1
```