

**1. ШУМЯЩИЕ МЕДВЕЖАТА**

10 очков

10 секунд

В каждом издании газеты Postimees печатают одну часть комикса "Шумящие медвежата". К сожалению иногда в редакции случаются неполадки и бывает, что одну картинку печатают несколько раз. Вы, являясь поклонником медвежат, хотели бы знать, сколько новых (ранее не изданных) комиксов вы получили в прошлом году. К счастью у вас сохранились WWW-ссылки комиксов всего года. Все ссылки вида

`http://www.postimees.ee/koomiks/karudXXXX.jpg`,

где XXXX дата рисования (не издания) комикса. Так как известно, что Urmas Nemvalts (автор комикса) никогда не рисует двух картинок за день, достаточно написать программу, которая считает число различных дат в этом списке.

Входные данные. На первой строке текстового файла KARUD.SIS число изданий газеты N ( $N \leq 366$ ) и на каждой из N следующих строк URL одного комикса в указанном виде.

Выходные данные. В текстовый файл KARUD.VAL вывести одно целое число – число разных комиксов.

Пример.

KARUD.SIS

KARUD.VAL

5

4

`http://www.postimees.ee/koomiks/karud0101.jpg``http://www.postimees.ee/koomiks/karud0103.jpg``http://www.postimees.ee/koomiks/karud0104.jpg``http://www.postimees.ee/koomiks/karud0103.jpg``http://www.postimees.ee/koomiks/karud0106.jpg`**2. ПЕРЕСЕЧЕНИЕ ОТРЕЗКОВ**

15 очков

10 секунд

На плоскости дан отрезок и за тем ещё N отрезков, которые могут пересекаться с первым отрезком. Написать программу, которая считает, на сколько частей те N отрезков разделяют (своими точками пересечения) первый отрезок.

Входные данные. На первой строке текстового файла LOIK.SIS координаты концов первого отрезка в виде

$$x_1 \ y_1 \ x_2 \ y_2,$$

где все координаты целые числа, абсолютная величина которых не превышает 100. На следующей строке файла целое число N ( $1 \leq N \leq 50$ ) и на следующих N строках координаты ещё N отрезков в том же формате.

Выходные данные. В текстовый файл LOIK.VAL вывести одно целое число – искомое число частей первого отрезка.

Пример.

LOIK.SIS

LOIK.VAL

2 2 5 2

3

3

3 1 5 3

6 3 8 1

2 1 4 3

**3. ФИЛЬТР ПАКЕТОВ**

20 очков

10 секунд

Каждому компьютеру, подключенному к Интернет, присвоен 4-байтовый адрес, который записают в виде  $x.x.x.x$ . Также имеет 4-байтовый адрес каждая подсеть, подключенная к Интернет. Чтобы правильно регулировать движение данных, надо знать, какие компьютеры принадлежат каждой подсети. Для этого каждой подсети кроме адреса присвоена 4-байтовая т.н. маска подсети, которую записают в таком же виде как адреса. Маска указывает, какая часть адреса компьютера должна совпадать с адресом подсети, которой этот компьютер принадлежит – в двоичных представлениях адресов компьютера и подсети должны совпадать биты на позициях, в которых в маске встречается значение 1. Например, если адрес некоторой подсети 192.168.0.0 и её маска 255.255.255.252, то компьютеры с адресами 192.168.0.1 и 192.168.0.2 принадлежат этой подсети, а компьютер с адресом 192.168.0.4 нет (см. также таблицу).

десятичный вид	двоичный вид
192.168.0.0	11000000.10101000.00000000.00000000
255.255.255.252	11111111.11111111.11111111.11111100
192.168.0.1	11000000.10101000.00000000.00000001
192.168.0.2	11000000.10101000.00000000.00000010
192.168.0.4	11000000.10101000.00000000.00000100

Данные в Интернете передаются пакетами. Пакет состоит из заголовка и тела. Заголовок содержит разные данные, нужные для доставки пакета: тип пакета, данные отправителя и получателя, длина данных и т.д. Тело содержит собственно передаваемые данные. Задача фильтра следить за движением в сети и выводить своды о типах и направлениях пакетов.

Входные данные. На первой строке текстового файла `FILTER.SIS` адрес и маска исследуемой подсети. На второй строке число анализированных пакетов  $N$  ( $N \leq 50000$ ) и на каждой из следующих  $N$  строк данные одного пакета: тип (или протокол) пакета (1-байтовое значение), адрес отправителя, адрес получателя, длина тела пакета в байтах (2-байтовое значение). Все числа даны в десятичной системе.

Выходные данные. В текстовый файл `FILTER.VAL` вывести для каждого протокола (вида пакетов), встречающегося во входных данных, одну строку вида

P KP KB SP SB VP VB TP TB,

где P протокол и следующие данные о пакетах данного типа: KP и KB число пакетов и байтов местного, SP и SB входящего, VP и VB выходящего, а TP и TB транзитного движения. Местными считаем пакеты, отправитель и получатель которых оба из данной подсети; входящими пакеты, отправитель которых вне, а получатель из данной подсети; выходящими пакеты, отправитель которых из, а получатель вне данной подсети; транзитные пакеты, отправитель и получатель которых оба вне данной подсети. Число байтов – сумма длин тел всех пакетов. Результаты вывести в возрастающем порядке номеров протоколов.

<u>Пример.</u>	<code>FILTER.SIS</code>	<code>FILTER.VAL</code>
	192.168.0.0 255.255.255.252	3 2 130 1 20 1 120 0 0
	5	6 0 0 1 40 0 0 0 0
	3 192.168.0.1 192.168.0.2 120	
	3 192.168.0.2 192.168.0.1 10	
	3 192.168.0.1 192.168.0.4 120	
	3 192.168.0.4 192.168.0.1 20	
	6 192.168.0.4 192.168.0.2 40	

**4. ГРАВИТРИПС**

25 очков

10 секунд

Гравитрипс настольная игра для двух игроков, в которую играют чёрными и белыми фишками на вертикально расположенной прямоугольной доске из  $N \times M$  клеток. Клетки составляют  $N$  столбцов по  $M$  клеток. В начале игры все клетки пустые и игроки начинают расставлять свои фишки по клеткам с целью добиться цепочки своего цвета длиной  $K$  по вертикали, горизонтали или диагонали.

На каждом ходу игрок запускает одну фишку из верхнего конца некоторого столбца. Фишка падает вниз по столбцу и останавливается в самой нижней свободной клетке. Например, в положении А, если белый ходит в столбец 2, результатом является положение В, а если в столбец 3, то положение С. Иначе ходить нельзя, так как в столбцах 1 и 4 уже свободных мест нет.

V..M	V..M	V.VM
M.MV	M.MV	M.MV
V.VM	VVVM	V.VM
Положение А	Положение В	Положение С

Написать программу, которая проверяет является ли данная последовательность положений корректной игрой по следующим правилам:

- первым ходит белый игрок;
- ходят строго по порядку по одной фишке на ход;
- если один из игроков добивается цепочки из  $K$  фишек по горизонтали, вертикали или диагонали, или если все клетки наполнятся фишками, игра сразу же кончится и больше никому ходить нельзя.

**Входные данные.** На первой строке текстового файла GRAVI.SIS 4 целых числа:  $N$ ,  $M$ ,  $K$  и  $X$ .  $N$  ( $3 \leq N \leq 15$ ) число столбцов и  $M$  ( $3 \leq M \leq 15$ ) число строк игрового поля.  $K$  ( $3 \leq K \leq \min(N, M)$ ) длина выигрышной цепочки.  $X$  ( $1 \leq X \leq N \times M$ ) число положений, описанных во входном файле. На каждой из следующих  $X$  строк по одному положению в порядке клеток по столбцам слева направо и сверху вниз (см. пример), где 'V' обозначает белую и 'M' черную фишку, а '.' пустую клетку. Первое указанное положение не обязательно начальное положение игры (пустое поле), но его корректность надо проверить тем не менее.

**Выходные данные.** В текстовый файл GRAVI.VAL вывести одно целое число – номер первого некорректного положения. Если все положения (и ходы, переводящиеся из предыдущего положения в следующее) соответствуют правилам, то вывести  $X+1$ .

**Пример.**

GRAVI.SIS	GRAVI.VAL
4 3 3 3	3
VMV...MVMVM	
VMV..V.MVMVM	
VMV...VMVMVM	

(В входном файле описаны те же положения, что на рисунке. Третье положение является нарушением правил, так как а) во втором положении белые уже выиграли и игра тем окончена; и б) вместо хода с чёрной фишкой переместили белую фишку.)

**5. КОВЕ**

30 очков

10 секунд

КОВЕ (треугольные шашки) настольная игра для одного игрока, в которую играют на треугольной доске, клетки которого пронумерованы по строкам сверху вниз и слева направо (см. рисунок).

```

      1
     2 3
    4 5 6
   7 8 9 10
  11 12 13 14 15

```

В начале игры в произвольно выбранные клетки расставляют некоторые фишки и дальше на каждом ходу надо с некоторой фишкой перепрыгнуть через соседние по диагонали фишки (например, фишкой из клетки 4 можно перепрыгнуть через фишки в клетках 2, 8 и 7 и "приземлиться" соответственно в клетках 1, 13 и 11). Прыгнуть можно только в случае, если в соседней клетке есть фишка, а клетка за ней пустая. Фишку, через которую перепрыгнули, удаляют с доски. Цель игры – добиться, чтоб на доске осталась только одна фишка.

Входные данные. На первой строке текстового файла КОВЕ.SIS число строк доски  $N$  ( $3 \leq N \leq 5$ ) и число фишек на доске  $M$  ( $2 \leq M$ ). На второй строке файла  $M$  разделённых пробелами целых чисел – номера клеток, содержащие фишки в начальном положении. Номера клеток указаны в возрастающем порядке.

Выходные данные. Если цель достижима (если возможно, что наконец на доске остаётся только одна фишка), то вывести на первую строку текстового файла КОВЕ.VAL 'JAN' и на следующие  $M-1$  строки ходы в порядке их исполнения, каждую на отдельную строку. Ход из клетки  $v_1$  в клетку  $v_2$  вывести в виде ' $v_1 v_2$ '. Если цель недостижима, то вывести на первую строку файла 'EI' и на вторую строку минимальное число фишек, оставшихся на доске.

<u>Пример.</u>	КОВЕ.SIS	КОВЕ.VAL
	3 3	JAN
	2 3 4	4 1
		1 6

<u>Пример.</u>	КОВЕ.SIS	КОВЕ.VAL
	3 2	EI
	2 3	2