

## Задача А. Радиоактивные шары

Входной файл:	<i>нет</i>
Выходной файл:	<i>нет</i>
Ограничение по времени:	1 секунда
Ограничение памяти:	64 мегабайта

В лабораторию на исследование поступило  $N$  шаров,  $K$  из которых радиоактивны ( $1 \leq K \leq N < 25$ ). В лаборатории имеется аппарат для проверки на радиоактивность. В него можно загрузить несколько шаров, после чего аппарат сообщает, есть ли в нем хотя бы один радиоактивный шар.

Лаборант Миша хочет за минимальное количество загрузок аппарата определить, какие из шаров радиоактивны. Помогите ему. Так как Миша боится ошибиться и быть уволенным, он хочет, чтобы результат, сообщенный им в отчете, был наверняка правильным.

В этой задаче вам требуется написать программу, которая будет взаимодействовать с модулем. Программа не должна пытаться читать либо записывать информацию в какие-либо файлы на диске.

### Описание взаимодействия с модулем

Вам доступны следующие процедуры и функции:

**Init** — инициализация, возвращает количество шаров и количество радиоактивных среди них.

**IsRadioactive** — проверка на радиоактивность, принимает в качестве параметра массив нулей и единиц, возвращает есть ли среди шаров, номерам которых соответствуют единицы, радиоактивные. Шары нумеруются от 0 до  $N - 1$ .

**Finish** — используется для вывода ответа, принимает в качестве параметра массив из нулей и единиц, в котором единицами должны быть помечены шары, которые радиоактивны.

Ваша программа должна вызвать процедуру **Init** для получения параметров задачи, после этого с использованием функции **IsRadioactive** выяснить, какие шары являются радиоактивными, сообщить об этом модулю с использованием процедуры **Finish** и завершить работу. Вам будут начислены баллы, в зависимости от количества вызовов функции **IsRadioactive**.

На время тура вам доступен демонстрационный модуль. Этот модуль отличается от того, который будет использоваться при тестировании вашей программы. Он читает из файла “balls.in” числа  $N$ ,  $K$  и затем  $N$  чисел, каждое из которых равно 0 или 1, задающих, какие шары являются радиоактивными. Среди них должно быть рав-

но  $K$  единиц. После диалога с программой, демонстрационный модуль создает файл “balls.out”, который содержит информацию о том, правильно ли были определены радиоактивные шары, и если да, то сколько запросов потребовалось для определения. Конечно, демонстрационный модуль не учитывает тот факт, что Миша должен представить наверняка правильный результат.

### Инструкция для пишущих на паскале:

Вам доступны два файла: “balls.ppw” и “balls.ow”. Данные файлы представляют собой реализацию модуля, интерфейсная часть приведена ниже:

```
procedure Init(var n, k: longint);

function IsRadioactive(mask: array of longint): boolean;

procedure Finish(mask: array of longint);
```

### Инструкция для пишущих на Си и Си++:

Вам доступны два файла: “balls.h” и “balls.o”. Данные файлы представляют собой реализацию модуля, интерфейсная часть приведена ниже:

```
void Init(int* n, int* k);

int IsRadioactive(int* mask);

void Finish(int* mask);
```

## Задача В. Шедевр

Входной файл:	masterpiece.in
Выходной файл:	masterpiece.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Новый футуристический шедевр, задуманный гениальным художником М. Калевичем, представляет собой прямоугольник, разбитый на  $N \times M$  клеток ( $N$  строк по  $M$  клеток). Знаменитый художник составил эскиз, на котором для каждой клетки решил, каким цветом она должна быть в соответствии с великим замыслом. Так как художник недавно примкнул к движению минималистов в живописи, он использует только

два цвета — черный и белый. Калевичу уже немало лет, поэтому за рабочий день он может сделать только один мазок — опустить кисть в одну из двух банок с краской и провести некоторую горизонтальную полосу высоты 1 по какой-то еще незакрашенной части холста. Перекрашивать холст нельзя. Калевич не хочет работать более  $K$  дней, так как уже наметил для себя творческий отдых. Какое наименьшее количество клеток окажется покрашенными неправильно? Заметим, что исходно холст имеет серый цвет.

### Описание входного файла

В первой строке входного файла записано три целых числа  $N$ ,  $M$  и  $K$  ( $1 \leq N \leq 64$ ,  $1 \leq M \leq 64$ ,  $1 \leq K \leq 4096$ ). Далее в  $N$  строках содержится описание эскиза. Каждая строка содержит по  $M$  символов. Символ '0' обозначает, что соответствующая клетка черная, '1' — белая.

### Описание выходного файла

Выведите наименьшее количество клеток, которые могут остаться покрашенными неправильно после не более чем  $K$  дней работы великого мастера. Далее в  $N$  строках выведите результат работы художника. Серый цвет следует обозначать символом '2'.

### Пример

masterpiece.in	masterpiece.out
3 5 3	2
00000	00000
00000	00000
00110	00000

## Задача С. Прекрасная принцесса

Входной файл:	princess.in
Выходной файл:	princess.out
Ограничение по времени:	2 секунды
Ограничение памяти:	64 мегабайта

Отважный искатель приключений Митя узнал, что злобные монстры похитили прекрасную принцессу и держат ее в таинственном лабиринте. Он решил ее спасти, тем более что накануне ему удалось достать содержащуюся в строжайшем секрете полную карту лабиринта. Лабиринт представляет собой прямоугольник  $m \times n$ , огороженный со всех сторон стенками, за исключением  $k$  выходов. Кроме того, некоторые клетки лабиринта также отделены друг от друга стенками. Самая странная особенность лабиринта — имеющиеся в нем таинственные образования, *дырки*. Дырки объединены в

циклические системы, и действуют таким образом, что провалившийся в  $i$ -ю дырку системы вылезает в  $i + 1$ -й (за исключением случая, когда дырка имела максимальный номер в системе — тогда провалившийся вылезает в самой первой дырке этой системы).

Некоторые клетки лабиринта охраняются злобными монстрами. Митя заранее узнал, сколько требуется для убийства и обезвреживания каждого из монстров:

- Для убийства разбойника требуется два патрона.
- Для убийства минотавра — четыре.
- На амазонку необходимо шесть.
- Варвару нужно целых восемь.
- А чтобы убить викинга, придется потратить целых десять патронов.
- Кроме того, в лабиринте водятся коровы. Их невозможно убить, но они и не опасны, поэтому на них не требуется тратить патроны.

Прилетев к месту операции на своем вертолете, Митя по неосторожности высадился в клетку с координатами  $(x, y)$  и отправил своего пилота вместе с вертолетом отдыхать в окружающий лабиринт город. Теперь ему необходимо рассчитать, как же добраться до принцессы и вывести ее из лабиринта, потратив при этом минимально возможное количество патронов (они ему еще понадобятся для спасения следующей принцессы из другого лабиринта).

Перемещаясь по лабиринту, Митя может ходить из клетки в соседнюю с ней по стороне, если эти две клетки не разделены стеной. Попадая на клетку с дыркой, Митя немедленно в нее проваливается, и вылезает в следующей дырке системы, после чего он может либо сходить в любую из четырех сторон, либо повторно провалиться в дырку. Естественно, что проходя по какой-то клетке, Митя убивает монстра, находящегося в ней, стреляя в него ровно необходимое количество патронов (при последующих проходах по этой клетке, конечно же, никаких патронов больше не требуется).

Напишите программу, которая рассчитывает минимальное требуемое количество патронов, а также один из возможных маршрутов для Мити, на прохождение которого требуется такое количество патронов. Маршрут должен удовлетворять правилам, описанным выше.

### Описание входного файла

В первой строке входного файла заданы два целых числа  $m$  и  $n$  ( $1 \leq m, n \leq 50$ ). Во второй строке содержится одно число  $k$  — количество выходов из лабиринта ( $1 \leq k \leq 2(m+n)$ ). В последующих  $k$  строках содержится описание выходов в виде  $x_i y_i d_i$ , где  $1 \leq x_i \leq m$ ,  $1 \leq y_i \leq n$  — координаты выхода, а  $d_i$  — одно из следующих четырех направлений:

- L — влево (уменьшить  $x$  на 1)

- R — вправо (увеличить  $x$  на 1)
- U — вверх (уменьшить  $y$  на 1)
- D — вниз (увеличить  $y$  на 1)

За описанием выходов следует описание монстров. Оно начинается со строки, содержащей число монстров  $z$  ( $0 \leq z \leq 20000$ ), за которой следуют  $z$  строк в формате  $x_j y_j T_j$ , где  $1 \leq x_j \leq m$ ,  $1 \leq y_j \leq n$  — координаты монстра, а  $T_j$  — его тип:

- R — разбойник
- MT — минотавр
- AM — амазонка
- BA — варвар
- BU — викинг
- K — королева

Все буквы в описании типа монстра — заглавные латинские. На одной и той же клетке может находиться несколько монстров. Монстры всегда охраняют ту и только ту клетку, на которой они находятся, и не перемещаются по лабиринту.

Далее следует описание стенок внутри лабиринта, начинающееся с числа стенок  $w$  ( $0 \leq w \leq 2mn$ ). За ней следуют  $w$  строк в формате, совпадающем с форматом описания выходов. Стенки являются двусторонними.

После описания стенок описываются дырки. В первой строке блока описания дырок содержится количество зацикленных систем  $l$  ( $0 \leq l \leq mn - 1$ ). Далее следуют  $l$  строк, каждая из которых начинается с числа  $p_i \geq 1$  — количества дырок в  $i$ -й системе, за которым следуют  $p_i$  пар чисел — координаты дырок  $i$ -й системы. Никакие две дырки не находятся на одной клетке.

В последних двух строках содержатся координаты Митиной высадки  $x$  и  $y$  ( $1 \leq x \leq m$ ,  $1 \leq y \leq n$ ), а также координаты клетки, в которой заточена принцесса  $x_p$  и  $y_p$  ( $1 \leq x_p \leq m$ ,  $1 \leq y_p \leq n$ ). Естественно, что если какая-то из этих двух клеток контролируется монстром, то Митя также будет вынужден с ним разобраться. Если Митя высаживается в клетке, содержащей дырку, он тут же в нее проваливается. Клетки с принцессой и с монстрами дырок не содержат.

### Описание выходного файла

Если требуемого маршрута не существует, в выходной файл необходимо вывести число  $-1$ . Иначе в первой строке должно быть выведено минимальное число патронов, которое придется потратить. За этой строкой должен следовать весь маршрут, пройденный Митей, начиная от клетки  $(x, y)$ , и до клетки выхода, удовлетворяющий описанным выше условиям на маршрут (в частности, если Митя попадает на клетку с дыркой, он вынужден немедленно в эту дырку провалиться). Никакая клетка не

должна встречаться два раза подряд в описании маршрута, даже если это ведущая сама в себя дырка. Кроме того, длина маршрута не должна превосходить  $50mn$ .

princess.in	princess.out
3 4	2
1	1 4
3 2 R	1 1
6	2 2
1 2 BU	1 4
1 2 BU	1 3
2 4 P	1 4
2 4 K	1 1
3 2 K	2 2
3 3 MT	1 4
7	2 4
1 1 R	3 4
2 1 D	2 3
3 2 L	2 1
2 3 U	3 1
2 3 L	3 2
2 3 D	
3 4 U	
2	
3 1 1 2 2 1 4	
3 2 1 3 4 2 3	
1 4	
1 3	