

**День 1****Общее описание задач**

<b>№</b>	<b>Название задачи</b>	<b>Ограничение на объем используемой памяти</b>	<b>Ограничение на время выполнения программы, секунды</b>	<b>Количество баллов присвоенных задаче</b>
1.	Абстрактное искусство	$\leq 256$ Кб	$\leq 1,500$	100
2.	Изображения	$\leq 256$ Кб	$\leq 0,002$	100
3.	Сообщения	$\leq 400$ Кб	$\leq 0,006$	100

*Примечание.* При равенстве общего числа очков, лучшим будет считаться решения с меньшим временем выполнения. В случае равенства и времени выполнения, лучшим будет считаться решение, использующее меньший объем памяти.

## Абстрактное искусство

Известный художник-абстракционист Эль-Абсурдо решил создать бессмертное произведение. По его мнению, оригинальность работы будет заключаться в том, что она будет образована только квадратами, но, в отличие от других картин такого типа, вершины каждого из рассматриваемых квадратов будут одного цвета.

Набрав несколько точек на холсте, Эль-Абсурдо раскрасил их, намереваясь нарисовать на них как можно больше квадратов, разумеется, с вершинами одного цвета. Эль-Абсурдо задумчиво прервал свой творческий процесс, намереваясь сначала выяснить максимально возможное количество таких квадратов.

Хорошо разбирающийся в информатике, Эль-Абсурдо ввел в исследование следующие обозначения.

Рассматриваются  $n$  точек, окрашенных в разные цвета. В учебных целях каждая точка  $i$  определяется декартовыми координатами  $(x_i, y_i)$ , а ее цвет задается целым числом  $c_i$  из закрытого интервала  $[1, k]$ . Требуется рассчитать максимально возможное количество квадратов  $P$ , имеющих вершины одного цвета.

**Задание.** Составьте программу, которая, зная декартовы координаты и цвет каждой из  $n$  точек, вычисляет максимальное количество квадратов, имеющих вершин одного цвета.

**Входные данные.** Первая строка стандартного ввода содержит целое число  $n$ . Каждая из следующих  $n$  строк стандартного ввода содержит целые числа  $x_i, y_i, c_i$ , разделенные пробелом. Строка  $i + 1$  стандартного ввода содержит координаты и цвет  $i$ -точки.

**Выходные данные.** Стандартный вывод должен содержать в одной строке целое число  $P$ .

**Ограничения.**  $n \leq 200$ .  $k \leq 200$ . Координаты  $x_i, y_i$  являются целыми числами,  $|x_i| \leq 10^6$ ;  $|y_i| \leq 10^6$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `arta.pas`, `arta.c` или `arta.cpp`.

### Пример 1.

Ввод	Вывод
9	2
1 1 1	
2 1 2	
3 1 1	
1 2 2	
2 2 1	
3 2 2	
1 3 1	
2 3 2	
3 3 1	

### Пример 2.

Ввод	Вывод
9	4
100 100 1	
200 100 1	
100 200 1	
200 200 1	
100 300 1	
200 300 1	
300 100 1	
300 200 1	
300 300 2	

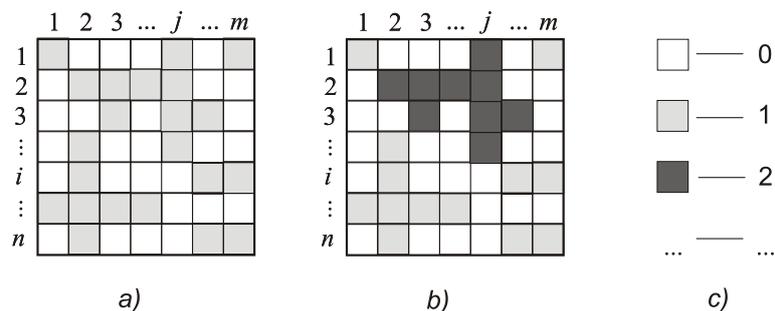
### Пример 3.

Ввод	Вывод
18	4
1 1 1	
2 1 1	
3 1 2	
4 1 2	
5 1 1	
6 1 1	
7 1 2	
8 1 2	
9 1 1	
1 2 1	
2 2 1	
3 2 2	
4 2 2	
5 2 1	
6 2 1	
7 2 2	
8 2 2	
9 2 1	

### Изображения

В памяти компьютера цветные изображения (см. рисунок) представляются в виде двумерного массива  $B = \parallel b_{ij} \parallel_{n \times m}$  содержащего  $n$  строк и  $m$  столбцов. Каждый элемент  $b_{ij}$  массива является натуральным числом указывающем в определенном коде цвет соответствующей микрозоны, например, белый ( $b_{ij} = 0$ ), черный ( $b_{ij} = 1$ ), красный ( $b_{ij} = 2$ ) и т.д. Для раскраски изображений цветные редакторы предлагают пользователю специальный инструмент  названный *Заливка*. Применение указанного инструмента моделирует процесс растекания краски из “баночки” в текущую микрозону, из текущей микрозоны в соседние микрозоны того же цвета и т.д. Очевидно, краска может растекаться из одной микрозоны в другую только в том случае если у них есть общая сторона.

**Задание.** Напишите программу для реализации инструмента *Заливка*.



Применение инструмента *Заливка*:

$a$  - исходное изображение;  $b$  - конечное изображение;  $c$  - кодировка цветов

**Входные данные.** Текстовый файл `IMAGINI.IN` содержит в первой строке натуральные числа  $n, m$  разделенные пробелом. Каждая из следующих  $n$  строк содержит по  $m$  чисел разделенных пробелами. Строка  $i+1$  входного файла содержит числа  $b_{i1}, b_{i2}, \dots, b_{im}$  исходного изображения. Последняя строка входного файла содержит три натуральных числа  $p, q, k$  разделенные пробелами. Числа  $p, q$  указывают координаты микрозоны к которой применяется инструмент *Заливка*, а число  $k$  – цвет краски из “баночки”.

**Выходные данные.** Текстовый файл `IMAGINI.OUT` должен содержать  $n$  строк, каждая из которых содержит по  $m$  чисел разделенных пробелами. Строка  $i$  выходного файла содержит числа  $b_{i1}, b_{i2}, \dots, b_{im}$  конечного изображения.

**Ограничения.**  $1 \leq n, m \leq 20, 0 \leq b_{ij}, k \leq 10, b_{ij} \neq k$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `IMAGINI.PAS, IMAGINI.C` или `IMAGINI.CPP`.

**Пример.** Для приведенного выше рисунка получаем:

IMAGINI.IN	IMAGINI.OUT
<pre>7 7 1 0 0 0 1 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 1 2 3 2</pre>	<pre>1 0 0 0 2 0 1 0 2 2 2 2 0 0 0 0 2 0 2 2 0 0 1 0 0 2 0 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 1</pre>

### Сообщения

Два друга передают друг другу сообщения. Чтобы быть уверенными что только адресат прочитает сообщение, они используют оригинальный метод. У друзей есть одинаковые дощечки с отверстиями. Для того, чтобы раскодировать сообщение, нужно наложить на текст дощечку так, чтобы в отверстия можно было видеть символы сообщения.

Дощечка с отверстиями представлена на рис. 1. Отправленный текст представлен на рис. 2. На рис. 3 представлена дощечка, наложенная на текст.

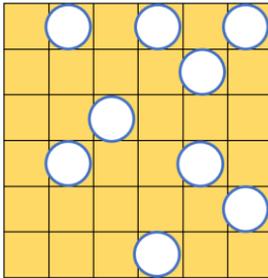


Рис. 1

m	a	i	f	o	o
o	a	c	s	s	a
a	p	t	n	r	f
p	o	a	e	d	o
t	a	v	a	e	a
f	s	r	t	u	t

Рис. 2

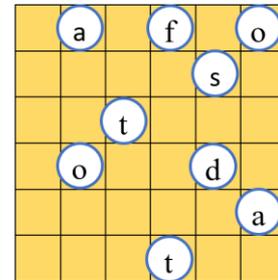


Рис. 3

Сообщение читается с верху вниз, слева на право. После того как были прочитаны все буквы, дощечка поворачивается по часовой стрелке на 90 градусов, читается следующий набор символов, и т.д. до полного оборота на 360 градусов.

**Задача.** Напишите программу, которая возвращает расшифрованное с помощью дощечки сообщение.

**Входные данные.** Стандартный ввод содержит на первой строке целое число  $N$  – количество строк и столбцов дощечки. На следующих  $N$  строках записаны  $N$  цифр, разделённые пробелами, которые описывают расположение отверстий на дощечке: 1 – нет отверстия, 0 – есть отверстие. На следующих  $N$  строках записаны  $N$  букв, разделённые пробелами, которые представляют закодированный текст.

**Выходные данные.** Стандартный выход содержит одну строку с расшифрованным сообщением, без пробелов.

**Пример.**

*Вход*

6
1 0 1 0 1 0
1 1 1 1 0 1
1 1 0 1 1 1
1 0 1 1 0 1
1 1 1 1 1 0
1 1 1 0 1 1
m a i f o o
o a c s s a
a p t n r f
p o a e d o
t a v a e a
f s r t u t

*Выход*

afostodatcanpovestiopreafrumoasafata
--------------------------------------

**Ограничения.**  $N \leq 100$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `mesaje.pas`, `mesaje.c` или `mesaje.cpp`.