

# Задания отборочного этапа олимпиады школьников «Ломоносов-2015» по информатике (10–11 классы) Первый тур

---

Задачи 1, 2, 3, 5, 6, 7 имеют вариативную постановку. Каждому участнику предлагался один из вариантов этих задач. Комментарии по этим вариантам приводятся ниже. Для приема решений использовалась система Ejudge.

## Задача 1. Упаковка числа

Рассмотрим способ упакованной записи чисел в двоичной системе. Идея способа состоит в том, чтобы одинаковые подряд идущие цифры, например 1111111, заменять на конструкцию, состоящую из повторяемой цифры (в примере – 1), открывающей скобки перед числом повторений – [, упакованной записи числа повторений в двоичной системе и закрывающей скобки – ]. Так число 111111100 в упакованной записи будет записано следующим образом: 1[1[1[10]]]0[10]

Найдите наименьшую по длине упакованную запись двоичного числа  
11100000000100000111111111110

Задача будет оценена в зависимости от верности упаковки и длины получившейся последовательности.

*Примечание:* в других вариантах этого задания предлагались разные длинные числа и основания систем счисления (двоичная или троичная).

## Задача 2. Последовательность

Последовательность чисел 38, 37, 36, 35, 39, 34, 33, 32, 31, 30, ... , 100 была получена выписыванием 500 первых натуральных чисел так, чтобы их записи римскими цифрами были отсортированы в обратном алфавитном порядке. Определите число, стоящее в последовательности на 114-м месте.

Указание: При решении задачи считайте, что в записи римскими цифрами цифра I может предшествовать только цифрам V и X, цифра V может предшествовать только цифре X, цифра X может предшествовать только цифрам L и C, цифра L может предшествовать только цифре C, цифра C может предшествовать только цифрам D и M. Для решения можно использовать электронные таблицы.

При сдаче решения в первой строке запишите ответ, в следующих строках приведите описание процесса получения ответа (это может быть - описание действий, программа и т.д.).

*Примечание:* в других вариантах предлагались другие последовательности.

### Задача 3. Битовые карты

Квадратная битовая карта  $A$  представляет собой матрицу размера  $n \times n$ , любой элемент  $a[i, j]$  которой может принимать значение либо 0, либо 1. В задаче будем использовать следующие простейшие операции над битовыми картами:

- $INV(A)$  – результатом является битовая карта  $B$ , в которой  $b[i, j] = 1$ , если  $a[i, j] = 0$ , и  $b[i, j] = 0$ , если  $a[i, j] = 1$ ;
- $MAX(A, B)$  – результатом является битовая карта  $C$ , в которой  $c[i, j] = \max(a[i, j], b[i, j])$ ;
- $MIN(A, B)$  – результатом является битовая карта  $C$ , в которой  $c[i, j] = \min(a[i, j], b[i, j])$ ;
- $TRANS(A)$  – результатом является битовая карта  $B$ , в которой  $b[i, j] = a[j, i]$ ;
- $MASK(A)$  – результатом является битовая карта  $B$ , в которой  $b[i, j] = 0$ , если  $i \leq n/2, j \geq n/2$ , и  $b[i, j] = a[i, j]$  при всех остальных  $i, j$ .

Из простейших операций можно составлять составные операции. Например, составная операция  $MIN(A, INV(TRANS(A)))$  применённая к следующей карте  $A$ :

1 0

1 1

даст в результате карту:

0 0

1 0

Запишите составную операцию, применение которой к произвольной неизвестной битовой карте  $X$  размера  $6 \times 6$  даст в результате битовую карту:

0 0 0 1 1 1

0 0 0 1 1 1

0 0 0 1 1 1

1 1 1 0 0 0

1 1 1 0 0 0

1 1 1 0 0 0

Запись искомой составной операции должна быть как можно меньшей длины.

*Примечание:* в других вариантах изменялся список простейших операций и итоговая битовая карта.

#### Задача 4. Упаковка числа - 2

Рассмотрим способ упакованной записи чисел в  $r$ -ичной системе. Идея способа состоит в том, чтобы одинаковые подряд идущие цифры, например 111111111, заменять на следующую конструкцию. Сначала идет повторяемая цифра (в примере – 1), затем открывающаяся скобка перед числом повторений [, затем упакованная запись числа повторений в  $r$ -ичной системе и, наконец, закрывающаяся скобка ]. Так число 111111111000 в упакованной записи в двоичной системе может быть записано следующим образом: 1[10[10]1]0[11].

Составьте программу, которая упаковывает число в  $r$ -ичной системе счисления. Длина упакованного числа должна быть минимальной.

В первой строке вводится основание системы счисления  $r$  ( $1 < r < 11$ ). Во второй строке находится число, записанное в  $r$ -ичной системе счисления, длина записи не превышает 1000 символов.

Пример:

| Ввод               | Вывод      |
|--------------------|------------|
| 2<br>1111111111000 | 1[1010]000 |

#### Задача 5. Пьюникод

Восстановите связное сообщение по тексту, который был получен после некоторого преобразования исходного текста:

xn--punycode-----ascii--

ujtaacbaerdf0bebiat2bbie8aged5ehg8cep7jivfj1bdsvfbbifadk3agcebcdglfdkfb1by2bcwd8ikh1a8cf2y4fmpub4he2w1a6t

*Примечание:* в других вариантах предлагался другой зашифрованный текст.

#### Задача 6. Уравняй числа

Результатом применения операции 1 к паре натуральных чисел  $(a, b)$  является пара натуральных чисел  $(a+1, 2 \times b)$ . Результатом применения операции 2 к паре натуральных чисел  $(a, b)$  является пара натуральных чисел  $(2 \times a, b+1)$ . Рассмотрим пару  $(3, 7)$ . Применяя к паре данные операции, можно получить пару равных чисел:  $(3, 7) \xrightarrow{1} (4, 14) \xrightarrow{2} (8, 15) \xrightarrow{2} (16, 16)$ .

Требуется за как можно меньшее число шагов при помощи операций 1 и 2 получить пару равных чисел из пары  $(63, 14)$ . Результат запишите в виде последовательности из единиц

и двоек (номеров операций) без пробелов. Если "уравнять" данную пару нельзя, укажите результат 0. Начиная со второй строки ответа, опишите, как было получено решение.

*Примечание:* в других вариантах предлагалась другая начальная пара чисел.

## Задача 7. КНБ

Альберт, Бертольд и Корвин играют втроем в игру "камень-ножницы-бумага". По счёту "три" все трое одновременно показывают рукой один из трёх возможных жестов, означающих камень (Rock), ножницы (Scissors) или бумагу (Paper). Считается, что камень побеждает ("затупляет") ножницы, ножницы побеждают ("разрезают") бумагу, а бумага побеждает ("заворачивает") камень.

На каждом кону разыгрывается одно очко. Если все трое показывают один и тот же жест, считается, что произошла ничья, и все получают по  $1/3$  очка. Если все трое показывают разные жесты, "победив" друг друга по кругу, тоже фиксируется ничья с таким же результатом. Если двое показывают одинаковый жест, и третий побеждает их обоих, то он получает одно очко, а остальные – ничего. Если двое, показав одинаковый жест, побеждают третьего, то они получают по  $1/2$  очка, а проигравший – ноль.

Протоколом игры называется последовательность трёхбуквенных слов, разделённых пробелами и/или переводами строки. Каждое слово протокола описывает исход одного кона, причём первая буква соответствует жесту Альберта, вторая – жесту Бертольда, третья – жесту Корвина. Камень кодируется буквой R, ножницы – буквой S, бумага – буквой P. Например, слово SRR означает, что Альберт показал "ножницы", а Бертольд и Корвин показали "камень" и, таким образом, получили по  $1/2$  очка.

Скачайте zip-архив с протоколом (предлагалась ссылка) и подсчитайте, сколько очков получил каждый из игроков в итоге всей игры, отражённой в протоколе. Ответ выдайте в виде трёх строк. В первой строке должен быть результат Альберта, во второй – результат Бертольда, в третьей – результат Корвина. Каждый результат запишите в виде: целая часть числа, пробел, а затем простая несократимая дробь в виде числителя и знаменателя, разделённых дробной чертой. Если дробная часть равна нулю, то не указывайте её в ответе. Если целая часть равна нулю, а дробная не равна, то не указывайте нулевую целую часть в ответе.

Пример протокола:

SRR PPP RSP PPS PSP RPP RPP SSR

Пример ответа:

2/3

3 1/6

4 1/6

*Примечание:* в других вариантах предлагались другие протоколы.

### Задача 8. Бармаглот-1

В вашем распоряжении оказался [эмулятор компьютера Barmaglot](#) (см. задачу 6 для 5-9 классов). Требуется написать для него программу, которая по заданному на ленте году выводит строку YES, если он високосный, и NO в противном случае. Год является високосным в двух случаях: либо он кратен 4, но при этом не кратен 100, либо кратен 400. Год не является високосным, если он не кратен 4, либо он кратен 100, но при этом не кратен 400.

### Задача 9. Бармаглот-2

В вашем распоряжении оказался [эмулятор компьютера Barmaglot](#) (см. задачу 6 для 5-9 классов). Требуется написать для него программу, которая находит и печатает максимальное число, среди записанных на ленте. Числа на ленте неотрицательные и разделены ровно одним пробелом.

### Задача 10. Дороги

Иван Васильевич является владельцем компании «В добрый путь», которая занимается ремонтом старых дорог и строительством новых. Правда, в последнее время дела идут плохо - никто не обращается к ним с просьбой что-то починить или построить.

Казалось бы, банкротства не избежать, но на днях появилось предложение, от которого Иван Васильевич не может отказаться. А именно, руководство города Байттаун объявило конкурс на реконструкцию дорог! Замечательная возможность поправить свое положение!

Всего в городе  $M = 1000$  дорог. Предприятие «Дураки и дороги» умеет делать  $N = 1000$  различных операций над каждой дорогой.

Вдохновленный этим предложением, Иван Васильевич вместе с главным инженером придумали  $K = 50$  планов реконструкции дорог. Архив с планами доступен по ссылке (предлагалась ссылка). Каждый план представляет собой матрицу размера  $M \times N$ , состоящей из 0 и 1, причем на пересечении  $i$ -й строки и  $j$ -го столбца стоит 1, если, согласно плану, нужно выполнить  $j$ -ю операцию над  $i$ -й дорогой, и 0 в противном случае.

Осталось только выбрать план, с которым нужно идти к заказчику. Но вот незадача - конкуренты из компании «Дорожная карта» умудрились украсть один из  $K$  вариантов и теперь собираются с ним идти на конкурс. Но программисты «Дураков и дорог» тоже не лыком шиты: они смогли получить ограниченный доступ к файлу на сервере конкурентов.

В связи с этим Иван Васильевич обращается к вам с просьбой помочь определить, какой именно вариант был украден, чтобы не идти с ним к заказчику.

Правда, вы не можете просто так посмотреть файл на сервере, иначе программисты «Дорожной карты» обнаружат подозрительную активность и перезагрузят сервер. Вместо этого вы можете узнать следующую информацию: в скольких ячейках какой-то подматрицы размера 5x5 лежит 1.

Решение должно выводить запросы на стандартный поток ввода. Каждый запрос выводится в отдельной строке. Существует два типа запросов:

- '?' i j' - запрос информации о содержимом файла число единиц в подматрицы размера 5x5 с левым верхнем углом в ячейке i j. В ответ на этот запрос на стандартный поток ввода поступит одно число - результат запроса. Запрашиваемая подматрица не должна выходить за границы исходной матрицы. Элементы нумеруются с нуля.
- '! ans' - программа угадала номер файла (ans). После этого запроса решение должно завершиться.

На каждый запрос '?' вводится число единиц в соответствующей подматрице.

После вывода каждого запроса решение должно выталкивать данные из буфера вывода с помощью следующих команд:

- На языке C: `fflush(stdout);`
- На языке Pascal: `flush(output);`
- На языке Python: `sys.stdout.flush();`
- На языке C++: `cout.flush();`
- На языке Java: `System.out.flush();`

Ниже приводится пример взаимодействия решения с сервером.

| Запросы | Ответы |
|---------|--------|
| ? 0 0   | 10     |
| ? 1 1   | 8      |
| ! 1     |        |

# Задания отборочного этапа олимпиады школьников «Ломоносов-2015» по информатике (10–11 классы) Второй тур

---

Задачи 1, 2, 5, 7 имеют вариативную постановку. Каждому участнику предлагался один из вариантов этих задач. Комментарии по этим вариантам приводятся ниже. Для приема решений использовалась система Ejudge.

## Задача 1. Сообщение

Восстановите связное сообщение по тексту, который был получен после некоторого преобразования исходного текста:

&#x0041; &#x0020; &#x0065; &#x0075; &#x006d; &  
; &#x0065; &#x0072; &#x0069; &#x0063;  
&#x0020; &#x0063; &#x0068; &#x0061; &#x0072; &  
; &#x0061; &#x0063; &#x0074; &#x0065;  
&#x0072; &#x0020; &#x0072; &#x0065; &#x0066; &  
; &#x0065; &#x0072; &#x0065; &#x0065;  
&#x0063; &#x0065; &#x0020; &#x0069; &#x0065; &  
; &#x0020; &#x0048; &#x0054; &#x004d;  
&#x004c; &#x0020; &#x0072; &#x0065; &#x0066; &  
; &#x0065; &#x0072; &#x0073; &#x0020;  
&#x0074; &#x006f; &#x0020; &#x0061; &#x0020; &  
; &#x0063; &#x0068; &#x0061; &#x0072;  
&#x0061; &#x0063; &#x0074; &#x0065; &#x0072; &  
; &#x0020; &#x0062; &#x0079; &#x0020;  
&#x0069; &#x0074; &#x0073; &#x0020; &#x0055; &  
; &#x0065; &#x0069; &#x0076; &#x0065;  
&#x0072; &#x0073; &#x0061; &#x006c; &#x0020; &  
; &#x0043; &#x0068; &#x0061; &#x0072;  
&#x0061; &#x0063; &#x0074; &#x0065; &#x0072; &  
; &#x0020; &#x0053; &#x0065; &#x0074;  
&#x002f; &#x0055; &#x0065; &#x0069; &#x0063; &  
; &#x006f; &#x0064; &#x0065; &#x0020;  
&#x0063; &#x006f; &#x0064; &#x0065; &#x0020; &  
; &#x0070; &#x006f; &#x0069; &#x0065;  
&#x0074; &#x002e;

*Примечание:* в других вариантах предлагались другие тексты.

## Задача 2. Последовательность

Последовательность чисел 2303, 511, 3311, 1519, 2527, 735, 1743, 2751, 959, 1967, ... , 1792 была получена следующим образом. Сначала были выписаны 500 первых натуральных чисел, кратных 7, по возрастанию (7, 14, ... 3500). Затем все числа были переведены в шестнадцатеричную систему, причём каждое число было записано тремя цифрами (007, 00E, ..., DAC). Затем каждая запись числа была преобразована в строку и символы в этой строке были записаны в обратном порядке (700, E00, ..., CAD). После чего полученные строки были отсортированы в порядке, обратном алфавитному (FF8, FF1, ..., 007). Затем строки были обратно преобразованы в исходные числа (2303, 511, ..., 1792). Определите номер места, на котором стоит в последовательности число 1414 (места нумеруются с 1).

Указание: Для решения можно использовать электронные таблицы.

При сдаче задачи в первой строке запишите ответ, в следующих строках приведите описание процесса получения ответа (это может быть - описание действий, программа и т.д.).

Примечание: в других вариантах предлагались другие последовательности чисел.

## Задача 3. Ёлочка

Перед Новым годом в Интернете появился [сервис «Ёлочка»](#). С его помощью можно завести себе web-страничку, на которой будет жить виртуальное дерево. Сначала высота ёлочки – 100 сантиметров. Каждые последующие 20 сантиметров зарабатываются просмотром странички в соответствии с последовательностью Фибоначчи. Т.е. для роста с 1 метра до 120 сантиметров нужен 1 просмотр, с 120 до 140 - 2 просмотра, с 140 до 160 потребуется 3 просмотра, с 160 до 180 уже 5 просмотров, а с 180 до 200 целых 8 просмотров. Таким образом, в результате  $1 + 2 + 3 + 5 + 8 = 19$  просмотров ваша ёлочка выросла до 2 метров. Ёлочка дискретная, поэтому её рост всегда кратен 20 см.

Вы поместили ссылку на ёлку в своей социальной сети. Некоторые ваши друзья посетили страничку, а некоторые поместили ссылку у себя (кстати, размещение ссылки на другой странице приравнивается к 5 просмотрам). 31 декабря рано утром (7:00:00) страничка была заблокирована (появился пароль), а к вам на почту пришел журнал посещений вашего дерева. Нужно вычислить, какой высоты стала ёлочка, и тогда страница вновь станет доступна (паролем является высота ёлочки в сантиметрах).

Модель роста ёлки, описанная в задаче, может не совпадать с моделью реального мира.

Формат журнала посещения:

- 1) время посещения страницы (2014-12-28 12:51:00)
- 2) тип действия: просмотр W, размещение на другой страничке S

Пример ввода:

2014-12-28 16:15:14 W



2014-12-28 18:37:01 S

2014-12-29 12:43:42 W

2014-12-30 14:05:18 W

2014-12-30 14:07:12 S

2014-12-30 19:02:36 W

Ответ к примеру:

180

Требуется написать программу, которая по логу определяет рост ёлочки в сантиметрах. Ввод-вывод осуществляется при помощи стандартных потоков или из/в файлы input.txt/output.txt.

#### Задача 4. Сравнение чисел

Рассмотрим способ упакованной записи чисел в двоичной системе. Идея способа состоит в том, чтобы некоторые одинаковые подряд идущие цифры заменять на конструкцию, состоящую из повторяемой цифры, открывающей скобки перед числом повторений - [, упакованной записи числа повторений в двоичной системе и закрывающей скобки - ].

Так все следующие записи соответствуют упакованной версии числа 111111100:

111111100

1[10]1[10]1[10]100

11111110[10]

1[111]00

1[111]0[10]

1[1[11]]00

1[1[11]]0[10]

1[1[1[10]]]00

1[1[1[10]]]0[10]

Требуется написать программу, сравнивающую два числа в упакованном виде. Числа даны на стандартном потоке ввода каждый в своей строке без ведущих нулей.

Гарантируется, что изначальные (распакованные) числа больше нуля и не превосходят  $2^{100000}$ .

В стандартный поток вывода необходимо напечатать <, если первое число меньше второго; =, если числа равны и >, если первое число больше второго.

Примеры:

| Ввод                          | Вывод |
|-------------------------------|-------|
| 111111100<br>1[1[1[10]]]0[10] | =     |
| 11<br>10                      | >     |
| 1[10]<br>101                  | <     |

## Задача 5. Найди питона

Представим, что последовательность из  $N$  десятичных цифр описывает питона длины  $N$ . Для описания питона будем использовать первые  $N$  цифр последовательности 012345678910111213141516171819202..., образованной выписыванием подряд цифр из записи неотрицательных натуральных чисел в порядке возрастания. 0 - первая цифра последовательности - соответствует голове питона,  $N$ -ная - хвосту.

На плоскости питон всегда сворачивается в спираль специальным образом. Голова свернувшегося питона всегда находится у центра спирали. Туловище свернувшегося питона следует либо по ходу часовой стрелки, если рассматривать его от головы к хвосту, либо против хода часовой стрелки. На рисунке показан свернувшийся питон 012345678910 длины 12. Рядом изображена схема, как свернулся этот питон.

```
.....
■9876■ ■
■1105■ ■
■0234■ ■
.....
```

Все питоны всегда сворачиваются по такой схеме плотно, без пропусков. Голова питона на рисунке справа направлена вправо. Также допускается, что питон сворачивается так, что его голова направлена вверх, влево или вниз.

Ваша задача: на заданном матрицам  $M$ , состоящей из  $K$  строк длиной  $K$  десятичных цифр ( $K > N$ ), найти свернувшегося в спираль питона  $P$ , имеющего самую большую длину среди всех питонов, находящихся на матрице, и выдать в результате координаты головы питона (номер строки, номер столбца) и длину питона. Учтите, что питоны в матрице могут пересекаться.

Пример:

$K$ : 7

$M$ :

1111210

1023430

1110510

1987640

1812700

0703000

0654000

Результат:

(3, 4)

20

Вам дается три файла с питонами. Требуется получить ответ для каждого из них по две строки, соответственно (всего 6 строк). После 6-й строки опишите, как был получен ответ (это может быть текст программы или словесное описание).

*Примечание:* в разных вариантах предлагались разные файлы с питонами.

### Задача 6. Уравняй числа

Результатом применения операции 1 к паре натуральных чисел  $(a, b)$  является пара натуральных чисел  $(a+1, 2 \times b)$ . Результатом применения операции 2 к паре натуральных чисел  $(a, b)$  является пара натуральных чисел  $(2 \times a, b+1)$ . Рассмотрим пару  $(3, 7)$ . Применяя к паре данные операции, можно получить пару равных чисел:  $(3, 7) \xrightarrow{1} (4, 14) \xrightarrow{2} (8, 15) \xrightarrow{2} (16, 16)$ .

Требуется за как можно меньшее число шагов с помощью операций 1 и 2 получить из пары  $(14, 20)$  пару равных чисел.

Результат запишите в виде последовательности из единиц и двоек (номеров операций) без пробелов. Если «уравнять» данную пару нельзя, укажите результат 0. Начиная со второй, строки опишите, как было получено решение.

### Задача 7. Рыцари круглого стола

$2n+1$  рыцарей Круглого стола всегда сидят за Круглым столом таким образом, что для каждого рыцаря справедливо свойство: рыцарь побеждает в поединке любого из  $n$  рыцарей, сидящих от него через 1, 3, 5,  $2n-1$  стульев, если считать стулья по часовой стрелке, и проигрывает в поединке любому из  $n$  рыцарей сидящих на оставшихся стульях.

Рыцари провели турнир, состоящий из нескольких поединков, в каждом из которых сражались по два рыцаря. Результаты они занесли в манускрипт. В первой строке манускрипта указано количество рыцарей. Во второй строке манускрипта указаны имена рыцарей, разделённые пробелами, перечисленные по порядку рассадки за Круглым столом по часовой стрелке, имя любого рыцаря состоит из не более 200 латинских букв. В

третьей строке манускрипта указано М – количество поединков в турнире. Оставшиеся М строк манускрипта содержат по одной паре имён рыцарей, разделённых пробелами. Каждая пара имён указывает рыцарей, сошедшихся в поединке. Не бывает поединков, в которых рыцарь сражается сам с собой.

По заданному манускрипту (дается ссылка на файл) определите итоговый счёт, учитывая, что за победу в поединке рыцарь получает 2 очка, а за поражение 0 очков.

Записывая ответ, упорядочите имена рыцарей по невозрастанию набранных в турнире очков. Имена рыцарей, набравших одинаковое количество очков, упорядочите по алфавиту. Каждое имя расположите на отдельной строке. После имени через пробел в той же строке укажите набранные рыцарем очки.

Пример манускрипта:

```
5
lancelot galahad percival arthur lamorak
5
lancelot galahad
lancelot percival
lancelot arthur
lancelot lamorak
percival galahad
```

Пример ответа:

```
lancelot 4
arthur 2
galahad 2
percival 2
lamorak 0
```

*Примечание:* в других вариантах предлагались другие файлы с манускриптами.

## Задача 8. Бармаглот-3

В вашем распоряжении оказался [эмулятор компьютера Barmaglot](#) (см. задачу 6 для 5-9 классов). Требуется написать для него программу, которая по заданной на ленте строке, состоящей из латинских символов, выводит её в перевернутом виде.

## Задача 9. Бармаглот-4

В вашем распоряжении оказался [эмулятор компьютера Barmaglot](#) (см. предыдущую задачу). Требуется написать для него программу, которая для каждого числа заданной последовательности целых чисел (больших 1) выводит, является ли оно простым или нет (один символ Y, если число простое и N если нет).

## Задача 10. Новые дороги

Иван Васильевич является владельцем компании «В добрый путь – 2», которая занимается ремонтом старых дорог и строительством новых. Правда, в последнее время дела идут плохо - никто не обращается к ним с просьбами что-то починить или построить.

Казалось бы, банкротства не избежать, но на днях появилось предложение, от которого Иван Васильевич не может отказаться. А именно, руководство города Байттаун объявило конкурс на реконструкцию дорог! Замечательная возможность поправить свое положение!

Всего в городе  $M = 1000$  дорог. Предприятие «Дураки и дороги» умеет делать  $N = 1000$  различных операций над каждой дорогой.

Вдохновленный этим предложением, Иван Васильевич вместе с главным инженером придумали  $K = 50$  планов реконструкции дорог. Архив с планами доступен по ссылке (ссылка прилагается). Каждый план представляет собой матрицу размера  $M \times N$ , состоящей из 0 и 1, причем на пересечении  $i$ -й строки и  $j$ -го столбца стоит 1, если, согласно плану, нужно выполнить  $j$ -ю операцию над  $i$ -й дорогой, и 0 в противном случае.

Осталось только выбрать план, с которым нужно идти к заказчику. Но вот незадача - конкуренты из компании «Дорожная карта» умудрились украсть один из  $K$  вариантов и теперь собираются с ним идти на конкурс. Но программисты «Дураков и дорог» тоже не лыком шиты - они смогли получить ограниченный доступ к файлу на сервере конкурентов.

В связи с этим Иван Васильевич обращается к вам с просьбой помочь определить, какой именно вариант был украден, чтобы не идти с ним к заказчику.

Правда, вы не можете просто так посмотреть файл на сервере, иначе программисты «Дорожной карты» обнаружат подозрительную активность и перезагрузят сервер. Вместо этого, вы можете узнать следующую информацию: чего больше в подматрице размера  $5 \times 5$  - нулей или единиц.

Решение должно выводить запросы на стандартный поток ввода. Каждый запрос выводится в отдельной строке. Существует два типа запросов:

- '? i j' - запрос информации о содержимом файла число единиц в подматрицы размера  $5 \times 5$  с левым верхним углом в ячейке  $i j$ . В ответ на этот запрос на стандартный поток ввода поступит одно число - результат запроса. Запрашиваемая

подматрица не должна выходить за границы исходной матрицы. Элементы нумеруются с нуля.

- '! ans' - программа угадала номер файла (ans). После этого запроса решение должно завершиться.

На каждый запрос '?' вводится "0", если нулей больше, и "1", если больше единиц в соответствующей подматрице.

После вывода каждого запроса решение должно выталкивать данные из буфера вывода с помощью следующих команд:

- На языке C: `fflush(stdout);`
- На языке Pascal: `flush(output);`
- На языке Python: `sys.stdout.flush();`
- На языке C++: `cout.flush();`
- На языке Java: `System.out.flush();`

Ниже приводится пример взаимодействия решения с сервером.

| Запросы | Ответы |
|---------|--------|
| ? 0 0   | 0      |
| ? 1 1   | 1      |
| ! 1     |        |