

Задача 1. Колонна

Полный балл:	100
Ограничение времени:	1 с
Ограничение памяти:	128М

Колонна дуболомов длиной L метров проходит мимо Урфина Джюса за T секунд. Помогите мудрому филину Гуамоко рассчитать, сколько времени понадобится этой колонне, чтобы пересечь мост длиной M метров. Время пересечения моста - это время между моментом, когда первый дуболом колонны ступил на мост, и моментом, когда последний дуболом колонны сошел с моста.

Формат входных данных

В первой строке входных данных записаны три натуральных числа L, T, M ($1 \leq L, T, M \leq 10^5$).

Формат результата

Выведите время, которое понадобится колонне дуболомов, чтобы пресечь мост. Выводимое число должно быть округлено ровно до трех знаков после десятичной точки.

Пример входных данных

1 2 3

Результат работы

8.000

Решение

Для того, чтобы найти время пересечения моста, необходимо поделить путь, пройденный колонной, на её скорость.

Для нахождения скорости колонны воспользуемся информацией о том, за какое время колонна проходит мимо Урфина Джюса. Посмотрим, какой путь проделывает начало колонны. Очевидно, что когда хвост колонны проходит мимо Урфина Джюса, начало колонны, находится на расстоянии L от Урфина, где L — заданная длина колонны. То есть, начало колонны проходит путь, равный L за время T , и скорость колонны $V = L/T$.

Теперь найдем путь, который проходит начало колонны при пересечении моста. Сначала передняя часть колонны движется от начала моста до его конца, проходя при этом путь M , равный длине моста. Затем колонна продолжает движение вперед, пока начало колонны не окажется на расстоянии L от конца моста. В этот момент конец колонны покидает мост, то есть колонна завершает движение по мосту. Таким образом, при пересечении колонной моста начало колонны проходит путь $M + L$. Время движения колонны через мост равно $(M + L) / V$.

При написании программы необходимо внимательно выбирать тип переменных для хранения заданных в условии величин. Так как предельные значения этих величин равны 10^5 , а сами величины — натуральные числа, то при использовании целочисленных переменных необходимо выбрать их тип и организовать вычисления так, чтобы не было переполнения.

В зависимости от языка программирования, необходимо также правильно выбрать операцию деления (целочисленное деление или деление чисел с плавающей точкой), а также выполнить приведение типа, чтобы получить в качестве ответа действительное, а не целое число.

По условию задачи, необходимо вывести ответ ровно с тремя знаками после запятой. Другой формат выводимого числа будет считаться ошибочным.

Ниже приведен один из возможных вариантов решения задачи на языке C.

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

```
#include <stdio.h>

long long L, T, M;

int main() {
    scanf("%lld %lld %lld", &L, &T, &M);

    double V = (double) L / T;
    double X = ((double)L + M) / V;
    printf("%.3lf\n", X);

    return 0;
}
```

Задача 2. Строй

Полный балл:	100
Ограничение времени:	1 с
Ограничение памяти:	128М

Урфин Джюс решил, что в его войске должны быть деревянные солдаты самого разного роста. Он приготовил список натуральных чисел и теперь хочет, чтобы для каждого числа из списка капрал Арум сказал ему, есть ли среди деревянных солдат дуболом такого роста.

Чтобы ответить Урфину, капрал Арум выстроил дуболомов по росту (от самого маленького до самого большого или, наоборот, от самого большого к самому маленькому). После этого он выбрал каких-то двух дуболомов и записал для каждого из них его номер в строю и рост. Также ему известно, сколько дуболомов стоит в строю, и что рост любых двух стоящих рядом дуболомов отличается на одну и ту же (возможно равную нулю) величину. То есть рост первого и второго отличается также, как рост второго и третьего, третьего и четвертого и так далее. Кроме того известно, что рост любого дуболома - натуральное число.

Напишите программу, которая по этим данным поможет капралу ответить для каждого числа из списка Урфина Джюса, есть ли в строю дуболом такого роста.

Формат входных данных

Первая строка входных данных содержит единственное натуральное число N ($2 \leq N \leq 10^6$) - количество дуболомов в строю.

Во второй строке записаны через пробел два натуральных числа $P1$ ($1 \leq P1 \leq N$) и $H1$ ($1 \leq H1 \leq 10^9$) - место в строю и рост первого из двух выбранных капралом дуболомов.

В третьей строке записаны через пробел два натуральных числа $P2$ ($1 \leq P2 \leq N$) и $H2$ ($1 \leq H2 \leq 10^9$) - место в строю и рост второго из двух выбранных капралом дуболомов.

В четвертой строке записано натуральное число T ($1 \leq T \leq 1000$) - количество чисел в списке Урфина Джюса.

В каждой из T последующих строк записано одно натуральное число M ($1 \leq M \leq 10^{18}$) - некоторое число из списка Урфина Джюса.

Формат результата

Для каждого числа из списка Урфина Джюса выведите в отдельную строку слово YES, если дуболом такого роста есть в строю дуболомов, или NO, если дуболома такого роста в строю нет.

Пример входных данных

```
6
2 6
5 15
3
8
9
10
```

Результат работы

```
NO
YES
NO
```

РЕШЕНИЕ

По описанию строя дуболомов понятно, что их рост образует арифметическую прогрессию. Зная номера двух элементов прогрессии и их значения, можно найти значение первого элемента прогрессии и разность прогрессии, решив систему двух линейных уравнений с двумя неизвестными. Обозначим через d разность арифметической прогрессии, а через X — значение первого элемента. Получаем систему двух уравнений с неизвестными X и d :

$$H1 = X + d * (P1 - 1);$$

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

$$H2 = X + d * (P2 - 1);$$

Из системы находим

$$d = (H2 - H1) / (P2 - P1);$$

$$X = H1 - d * (P1 - 1);$$

Теперь, если задано некоторое число N , можно определить принадлежит ли оно заданной арифметической прогрессии, и если да, то найти его номер в этой прогрессии.

Если число N принадлежит прогрессии, то $N = X + d * (P-1)$, где P — номер этого числа в прогрессии. Отсюда $P = (N - X) / d + 1$. Так как номер P — целое число, то для того, чтобы число N принадлежало прогрессии необходимо, чтобы разность $N - X$ без остатка делилась на d .

Если рост дуболомов в строю отличается на величину, равную нулю, то разность прогрессии $d = 0$, и вычислить P по формуле $(N - X) / d$ нельзя. Следовательно, случай, когда $d = 0$ необходимо обработать отдельно. Этот случай соответствует входным данным, когда $H1 = H2$, то есть рост двух выбранных дуболомов один и тот же. В этом случае, если заданное для проверки число равно $H1$, то ответ YES, то есть дуболом такого роста есть. Иначе ответ NO.

Если d не равно нулю, то сначала проверяем, делится ли разность $N - X$ на d без остатка. Если нет, то число N не принадлежит прогрессии и, следовательно, дуболома с таким ростом нет. Ответ NO.

Если же разность делится на d без остатка, то вычисляем номер значения N в прогрессии. Если этот номер меньше 1 или больше N , то ответ NO (дуболома с таким ростом в строю нет). Иначе ответ YES.

При реализации данного решения необходимо правильно выбрать тип переменных, чтобы избежать возможного переполнения при выполнении арифметических операций.

Какое либо переборное решение позволяет получить лишь неполный балл за задачу из-за достаточно больших ограничений - число дуболомов в строю $2 \leq N \leq 10^6$, количество чисел для проверки $1 \leq T \leq 1000$.

Ниже приводится один из возможных вариантов реализации решения на языке C.

```
#include <stdio.h>

long long N;
long long H1, P1, H2, P2;

int T;
long long H;

int main() {
    scanf("%lld", &N);
    scanf("%lld %lld %lld %lld", &P1, &H1, &P2, &H2);

    long long d = (H2 - H1) / (P2 - P1);
    long long X = H1 - d * (P1 - 1);

    scanf("%d", &T);

    while (T--) {
        scanf("%lld", &H);
```

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

```
    if (d == 0) {
        if (H == H1)
            printf("YES\n");
        else
            printf("NO\n");
        continue;
    }

    if ((H - X) % d != 0) {
        printf("NO\n"); continue;
    }

    long long P = (H - X) / d + 1;

    if (1 <= P && P <= N)
        printf("YES\n");
    else
        printf("NO\n");

}

return 0;
}
```

Задача 3. Единицы

Полный балл:	100
Ограничение времени:	1 с
Ограничение памяти:	128М

Волшебник Изумрудного города Гудвин оказался очень суеверным. Он считает цифру 1 несчастливой. Гудвин пригласил к себе мудрого филина Гуамоку и хочет, чтобы тот ответил на вопрос, сколько целых чисел между двумя заданными целыми числами L и R включительно содержат в своей десятичной записи хотя бы одну цифру 1.

Формат входных данных

Единственная строка входных данных содержит два целых числа L и R ($1 \leq L \leq R \leq 10^{10}$).

Формат результата

Выведите единственное целое число - количество целых чисел в диапазоне от L до R включительно, в десятичной записи которых есть хотя бы одна единица.

Пример входных данных

1 100

Результат работы

20

Пример входных данных

32 95

Результат работы

6

Пример входных данных

147 2180

Результат работы

1305

РЕШЕНИЕ

Заполним несколько вспомогательных массивов, которые помогут получить решение задачи.

Пусть $w[i]$ — количество чисел, в которых ровно i цифр и нет не одной единицы. Тогда $w[1] = 8$, так как однозначных чисел, начиная с 1, всего 9, но так как мы считаем числа, в которых нет не одной единицы, то само число 1 мы должны отбросить, и у нас останется ровно 8 однозначных чисел, среди которых нет единицы.

$w[2] = 8 * 9$. Действительно, $w[2]$ — это количество двузначных чисел, в записи которых нет ни одной единицы. Поэтому на в качестве первой цифры такого двухзначного числа мы можем использовать цифры 2-9, то есть всего восемь цифр, а в качестве второй — все цифры, кроме единицы, то есть 9 цифр. Всего вариантов возможных чисел $8 * 9 = 72$. То есть в диапазоне от 10 до 99 включительно есть ровно 72 числа, в записи которых нет цифры 1.

Аналогично, $w[3] = 8*9*9$. В качестве первой цифры трехзначного числа можно использовать 8 цифр, в качестве второй — 9 и в качестве третьей — тоже 9. Всего получается $8 * 9 * 9 = 648$ чисел в диапазоне от 100 до 999 включительно, в записи которых нет ни одной цифры 1.

И так далее. Последнее значение, которое нас интересует, это $w[10]$, так как по условию задачи для границ диапазона $[L, R]$, внутри которого мы должны считать количество чисел, в

записи которых есть хотя бы одна единица, выполняется неравенство $1 \leq L \leq R \leq 10^{10}$. Очевидно, что массив $w[i]$ можно заполнить следующим циклом:

```
w[1] = 1;
for (int i = 2; i <= 10; i++) {
    w[i] = w[i-1] * 9;
}
```

Заполним теперь массив $sum[i]$ — количество чисел в диапазоне от 1 до 10^i-1 включительно, в записи которых нет ни одной единицы. Очевидно, что $sum[i]$ равно сумме значений $w[i]$ от 1 до i включительно. То есть массив $sum[i]$ легко заполняется циклом

```
sum[1] = w[1];
for (int i = 2; i <= 10; i++) {
    sum[i] = sum[i-1] + w[i];
}
```

Посчитаем также в массиве $pw[i]$ степени числа 10, которые нам пригодятся при вычислении результата. Вычисление всех трех массивов можно объединить в одном цикле.

```
w[1] = 8; sum[1] = 8; pw[1] = 10;
for (int i = 2; i <= 10; i++) {
    w[i] = w[i-1] * 9;
    sum[i] = sum[i-1] + w[i];
    pw[i] = pw[i-1] * 10;
}
```

Посчитаем теперь в массиве $c[i]$ количество чисел в диапазоне от 1 до 10^i-1 в записи которых есть хотя бы одна единица. $c[i] = \langle \text{количество всех чисел в диапазоне от 1 до } 10^i-1 \rangle - \langle \text{количество чисел в диапазоне от 1 до } 10^i-1, \text{ в которых нет ни одной единицы} \rangle$. То есть $c[i] = 10^i-1 - sum[i]$, и массив $c[i]$ можно заполнить следующим циклом

```
for (int i = 1; i <= 10; i++) {
    c[i] = pw[i]-1 - sum[i];
}
```

Теперь можно написать рекурсивную функцию `long long get(long long N)`, которая будет возвращать количество чисел из диапазона от 1 до N включительно, в записи которых есть хотя бы одна цифра 1.

Функция устроена следующим образом. Посмотрим на первую цифру числа N . Если это цифра 1, то функция сразу может вернуть ответ. Пусть, например, $N = 1353$. Тогда количество чисел в диапазоне от 1 до 1353, в записи которых есть хотя бы одна единица — это количество таких чисел в диапазоне от 1 до 999 (это значение посчитано и хранится в ячейке массива $c[3]$), плюс одно число 1000, и плюс еще 353 числа, начинающихся с 1001 и заканчивающихся числом 1353.

Если мы обозначим через val число, получающееся из N отбрасыванием первой цифры, а через n — количество цифр в числе N , то результат, который должна вернуть функция в случае, если первая цифра числа N единица, задается выражением $c[n-1] + 1 + val$.

Если первая цифра d числа N отлична от 1, то функция `get` должна вернуть значение

$$(d-1) * c[n-1] + pw[n-1] + get(val)$$

Теперь ответ на запрос $[L, R]$ равен `get(R) — get(L-1)`. Случай, когда $L = 1$ лучше обработать отдельно.

Ниже приводится один из вариантов реализации этого решения на языке C.

```
#include <stdio.h>
#include <string.h>

long long L, R;

long long w[15];
long long sum[15];
long long c[15];
long long pw[15];

long long get(long long N)
{
    if (N == 0) return 0;
    if (N < 10) return 1;

    char s[20];
    sprintf(s, "%lld", N);
    int n = strlen(s);

    long long ans = 0;

    long long val = 0;
    for (int i = 1; i < n; i++) val = val*10 + (int)(s[i] - '0');

    if (s[0] == '1') {
        ans += c[n-1] + 1;
        ans += val;
    } else {
        ans += c[n-1] + pw[n-1];
        int d = (int)(s[0] - '0');
        ans += (d-2) * c[n-1];
        ans += get(val);
    }

    return ans;
}

long long get(long long L, long long R)
{
    if (L == 1) return get(R);
    return get(R) - get(L-1);
}

int main() {

    w[1] = 8; sum[1] = 8; pw[1] = 10;
    for (int i = 2; i <= 10; i++) {
        w[i] = w[i-1] * 9;
        sum[i] = sum[i-1] + w[i];
        pw[i] = pw[i-1] * 10;
    }

    for (int i = 1; i <= 10; i++) {
        c[i] = pw[i]-1 - sum[i];
    }

    scanf("%lld %lld", &L, &R);
    printf("%lld\n", get(L, R));

    return 0;
}
```


Задача 4. Удивительная таблица

Полный балл:	100
Ограничение времени:	1 с
Ограничение памяти:	128М

Страшила мудрый, побывав у Волшебника Изумрудного города Гудвина, великого и ужасного, узнал много удивительных вещей. Например, он видел у Гудвина удивительную числовую таблицу.

Таблица была квадратной, имела размер $N \times N$ и содержала только целые положительные числа. Страшила выбрал какое-то число в таблице, обвел его в кружок и вычеркнул все числа, которые были в той же строке и в том же столбце, что и обведенное число. Затем он выбрал следующее незачеркнутое число, обвел его в кружок и вычеркнул оставшиеся незачеркнутые числа в той же строке и в том же столбце, что и второе обведенное число. И так далее, пока в таблице не осталось незачеркнутых и необведенных в кружок чисел. Затем Страшила сложил все обведенные в кружок числа и запомнил получившуюся сумму. Потом он решил повторить всю эту процедуру с выбором чисел и вычеркиваниями еще раз с самого начала, а потом еще раз, и еще раз и еще много раз. Каждый раз, какие бы числа не оставались обведенными в кружок в конце этой процедуры, их сумма всегда была одна и та же.

Страшила так впечатлила эта удивительная таблица, что он решил рассказать о ней своим друзьям - Элли, Тотошке, Льву и Железному дровосеку. Но так как мозги у Гудвина Страшила еще не получил, то по дороге к друзьям он забыл почти всю таблицу. Все, что помнит Страшила - это первый столбец таблицы целиком и еще по одному числу в каждом из оставшихся столбцов.

Теперь только вы можете помочь Страшילה, написав программу, которая восстановит всю таблицу по тем данными, которые он запомнил.

Формат входных данных

Первая строка сходных данных содержит натуральное число N ($1 \leq N \leq 10$) - размер таблицы.

Следующие N строк содержат каждая по N чисел, разделенных пробелами. Эти числа либо целые положительные в диапазоне от 1 до 100 - это те числа, которые запомнил Страшила, либо нули. Нули соответствуют числам таблицы, которые Страшила забыл.

Формат результата

Выведите N строк, в каждой из которых записаны через один пробел N целых положительных чисел. Выведенная таблица должна обладать следующим свойством - какие бы числа из этой таблицы не выбрать так, чтобы в каждой строке и каждом столбце было выбрано ровно одно число, их сумма всегда будет одним и тем же числом. Выводимая таблица должна содержать те числа, которые запомнил Страшила, в тех же самых позициях, что и во входной таблице. Входные данные будут такими, что искомая таблица существует.

Пример входных данных

```
3
45 37 0
61 0 0
82 0 96
```

Результат работы

```
45 37 59
61 53 75
82 74 96
```

РЕШЕНИЕ

Для того, чтобы заполнить таблицу требующимся в условии способом, заметим следующее свойство, которым обладают числа в таблице.

Возьмем, например, таблицу размером 5×5 . Рассмотрим первый и второй столбцы этой таблицы. Пусть они имеют вид:

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

	1	2	3	4	5
1	A		*		
2	B	F			
3	C			*	
4	D	X			
5	E				*

Здесь A, B, C, D, E, F — числа заданные в условии. X — число, которое мы хотим найти. Предположим, что из первого столбца было выбрано число B, из второго — число X, а из остальных столбцов некоторые числа из клеток, отмеченных *. Тогда из каждой строки и каждого столбца будет выбрано ровно одно число, и

$$B+X+S_{3-5} = S,$$

где S_{3-5} — это сумма чисел, выбранных в столбцах 3-5 (из клеток, отмеченных *), а S — сумма всех выбранных чисел. По условию задачи сумма S не меняется, как бы мы не выбирали числа по заданному в условии правилу.

Если теперь выбрать в первом столбце число D, во втором число F, а в остальных столбцах оставить те же числа, что там и были (это не нарушит правило выбора, при котором в каждой строке и каждом столбце должно стоять ровно одно число), получим:

$$D+F+S_{3-5} = S.$$

Из этих двух равенств следует, что $B+X = D+F$, и $X = D+F-B$.

Это свойство выполняется для любой клетки во втором столбце, так как клетку X мы выбрали произвольно. Следовательно мы можем заполнить второй столбец значениями:

1-я строка: $A+F-B$

3-я строка: $C+F-B$

4-я строка: $D+F-B$

5-я строка: $E+F-B$

Заполнив второй столбец, мы аналогично можем заполнить третий, затем, зная третий, заполним четвертый, и так далее — всю таблицу.

Программа на языке C, реализующая это решение, приводится ниже.

```
#include <stdio.h>

int N;
int a[100][100];

int main() {
    scanf("%d", &N);

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            scanf("%d", &a[i][j]);
        }
    }

    for (int j = 1; j < N; j++) {
        int pos = -1;
```

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

```
        for (int i = 0; i < N; i++) {
            if (a[i][j] != 0) {
                pos = i; break;
            }
        }
        for (int i = 0; i < N; i++) {
            if (i == pos) continue;
            a[i][j] = a[pos][j] + a[i][j-1] - a[pos][j-1];
        }
    }

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }

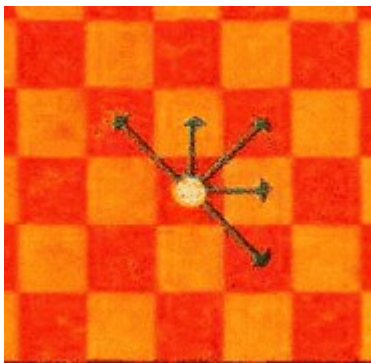
    return 0;
}
```

Задача 5. Игра в одну шашку

Полный балл:	100
Ограничение времени:	1 с
Ограничение памяти:	128М

Получив у Гудвина мозги, Страшила решил их проверить. Для этого он предложил сыграть Гудвину в следующую игру.

На шахматную доску размера $N \times N$ ставится одна единственная шашка. Игроки ходят по очереди. За один ход игрок может передвинуть шашку на одну клеточку только в направлениях, показанных на рисунке.



Выигрывает игрок, который поставит шашку в правый верхний угол шахматной доски. Страшила ходит первым. Зная размеры доски и координаты начального положения шашки, подскажите Страшиле, выиграет он или проиграет, если будет играть оптимально и он сам, и Гудвин, великий и ужасный.

Формат входных данных

Первая строка входных данных содержит натуральное число N ($1 \leq N \leq 10^9$) - размер доски, на которой происходит игра.

Следующая строка содержит натуральное число T ($1 \leq T \leq 100$) - количество партий, которые сыграли Страшила и Гудвин на этой доске. Каждая партия задается начальным положением шашки. Для каждой партии необходимо определить, кто выиграет при оптимальной игре обоих игроков - Страшила или Гудвин. В каждой партии первым всегда ходит Страшила.

В каждой из T последующих строк записаны через пробел два натуральных числа X и Y ($1 \leq X, Y \leq N$) - номер строки и столбца, в которых была расположена шашка в начале очередной партии. Строки доски нумеруются сверху вниз, начиная с 1. Столбцы доски нумеруются слева направо, начиная с 1. Левый нижний угол доски расположен в строке N и столбце 1. Правый верхний угол доски, где в конце игры должна оказаться шашка, расположен в строке 1 и столбце N . В начальной позиции шашка никогда не располагается в правом верхнем углу доски.

Формат результата

Для каждой заданной начальной позиции шашки из входных данных выведите в отдельную строку слово Scarecrow, если при оптимальной игре обоих игроков из этой начальной позиции выиграет Страшила. Если выиграет Гудвин, выведите слово Wizard.

Входные данные

```
8
3
1 7
2 7
1 6
```

Результат работы

```
Scarecrow
Scarecrow
Wizard
```

Пояснение к примеру

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

- Из клетки (1,7), расположенной в строке 1 столбце 7, Страшила сразу сделает ход в клетку (1,8) и выиграет.
- Аналогично, из клетки (2,7) Страшила сразу сделает ход в клетку (1,8) и тоже выиграет.
- Если партия начнется из клетки (1,6), то ход Страшилы приведет шашку либо в клетку (1,7), либо в клетку (2,7), и следующим ходом выиграет Гудвин.

РЕШЕНИЕ

Найдем в этой игре выигрышные-проигрышные клетки (позиции). Возьмем для определенности доску размером 5x5 клеток. Будем обозначать выигрышную клетку буквой В, а проигрышную — буквой П.

Клетка называется выигрышной, если игрок, оказавшийся в этой клетке, обязательно выиграет при правильной игре. Клетка называется проигрышной, если игрок, оказавшийся в этой клетке, обязательно проиграет при правильной игре соперника.

Очевидно, что правый верхний угол доски, клетка (1,5) - проигрышная клетка. Игрок, который должен ходить в этой клетке проиграл, так его соперник завершил игру и выиграл, поставив шашку в эту клетку.

	1	2	3	4	5
1					П
2					
3					
4					
5					

Клетки (1,4), (2,4), (2,5) — это выигрышные клетки, так как у игрока, чья очередь ходить из этих клеток есть ход, ведущий в клетку П и, следовательно, соперник, оказавшийся после хода игрока в клетке П, проиграет.

	1	2	3	4	5
1				В	П
2				В	В
3					
4					
5					

Клетки (1,3) и (3,5) — проигрышные, так как все ходы из этих клеток ведут в выигрышные клетки, и, следовательно, игрок, находящийся в клетке (1,3) или (3,5), неизбежно переведет игру в выигрышную клетку, в которой будет ходить его соперник и выиграет при правильной игре, так как клетка выигрышная.

	1	2	3	4	5
1			П	В	П
2				В	В
3					П
4					
5					

Теперь видно, что клетки (1,2), (2,3), (3,4), (4,5) выигрышные, так как из любой из этих клеток есть ход в проигрышную клетку.

**Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.**

	1	2	3	4	5
1		В	П	В	П
2			В	В	В
3				В	П
4					В
5					

Клетки (2,2) и (4,4) тоже выигрышные, так как из них есть ход в проигрышные клетки.

	1	2	3	4	5
1		В	П	В	П
2		В	В	В	В
3				В	П
4				В	В
5					

Клетки (1,1), (3,3), (5,5) — проигрышные, так как любой ход из этих клеток ведет в выигрышную клетку. Заполнив подобным образом всю таблицу, получим

	1	2	3	4	5
1	П	В	П	В	П
2	В	В	В	В	В
3	П	В	П	В	П
4	В	В	В	В	В
5	П	В	П	В	П

Аналогичная таблица для доски 6x6 выглядит следующим образом:

	1	2	3	4	5	6
1	В	П	В	П	В	П
2	В	В	В	В	В	В
3	В	П	В	П	В	П
4	В	В	В	В	В	В
5	В	П	В	П	В	П
6	В	В	В	В	В	В

Построив такую таблицу, можно легко определить, кто выиграет при правильной игре, если известна клетка, из которой начинается игра. Так как первым ходит Страшила, то он выиграет, если начальная клетка является выигрышной, то есть отмечена буквой В. Если начальная клетка отмечена буквой П, то страшила проигрывает, а выигрывает, соответственно, Гудвин.

Более того, чтобы определить, является клетка выигрышной или проигрышной, не обязательно строить и хранить всю таблицу, так как в расположении клеток В и П видна четкая закономерность. Если пронумеровать строки сверху вниз, а столбцы справа налево, то клетки П будут иметь нечетный номер строки и столбца:

Муниципальный этап всероссийской олимпиады школьников по программированию
Крым, 16 ноября 2019 г.

	6	5	4	3	2	1
1	В	П	В	П	В	П
2	В	В	В	В	В	В
3	В	П	В	П	В	П
4	В	В	В	В	В	В
5	В	П	В	П	В	П
6	В	В	В	В	В	В

Теперь решение задачи будет заключаться в следующем. Необходимо считать координаты начальной клетки и, правильно определив четность строки и столбца, выяснить, является эта клетка В или П. Если клетка В — вывести Scarecrow, если П - Wizard.

Возможный вариант реализации решения на языке С показан ниже.

```
#include <stdio.h>

int N;
int T;
int X, Y;

int main() {
    scanf("%d", &N);
    scanf("%d", &T);

    while (T-- > 0) {
        scanf("%d %d", &X, &Y);

        if (N % 2 == 0) {
            if (X % 2 != 0 && Y % 2 == 0)
                printf("Wizard\n");
            else
                printf("Scarecrow\n");
        } else {
            if (X % 2 != 0 && Y % 2 != 0)
                printf("Wizard\n");
            else
                printf("Scarecrow\n");
        }
    }

    return 0;
}
```