

Разбор задач

30 Районная олимпиада школьников

Красноярского края по информатике, 9-11 классы

Основной тур от 1 декабря 2016 г.

ID	Задача	Тема	Балл	%
1428	А. Окружность	Геометрия	100	29
1430	В. Максимальное число	Сортировка подсчетом	100	32
1427	С. Киберспорт	Сортировка структур	100	35
1424	Д. Нортландский шифр	Разбор строк, структуры данных	100	47
1423	Е. Красивая матрица	Бинарный поиск, моделирование	100	56

Задача А. Окружность

(Время: 1 сек. Память: 16 Мб Баллы: 100)

В центре системы координат находится окружность длины L .

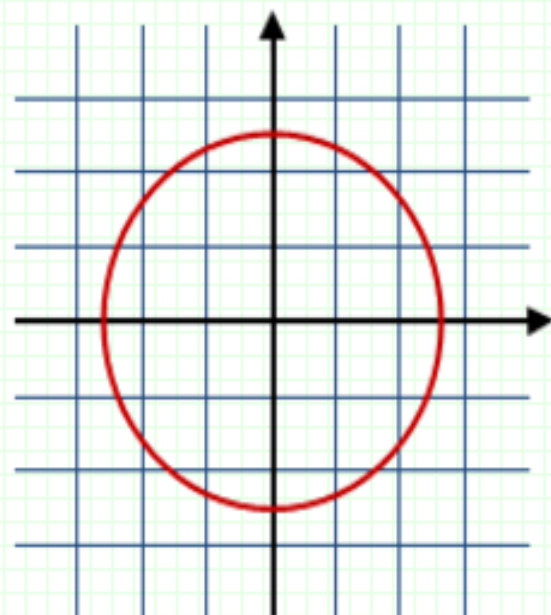
Требуется определить количество пересечений окружности с координатной сеткой.

Входные данные

Входной файл INPUT.TXT содержит натуральное число L ($L \leq 10^{18}$) – длину окружности.

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.



Примеры

№	INPUT.TXT	OUTPUT.TXT
1	5	4
2	17	20

Система оценки

Решения, работающие верно для $L \leq 25$, будут оцениваться в 30 баллов.

Задача А. Окружность

Зная длину окружности L , несложно вычислить ее радиус:

$$R = \frac{L}{2\pi}, \quad \pi \approx 3.141592653589793238462643$$

Можно заметить, что в каждой четверти окружность пересекает каждую вертикальную и горизонтальную линию сетки координат ровно один раз. Откуда получаем формулу, определяющую общее число таких пересечений:

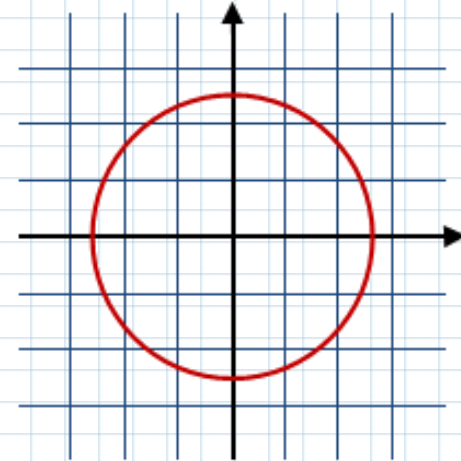
$$N = 4 \cdot (2 \cdot \lfloor R \rfloor + 1)$$

Формула верна, т.к. линия окружности никогда не проходит через целочисленные координаты. Это следует из того, что R - трансцендентное число, а подобное пересечение было бы возможно только при алгебраических значениях R .

Наибольшая проблема при решении данной задачи в том, что полученное значение R может быть очень близко к целому значению и стандартных типов `extended` (паскаль) и `long double` (C++) не хватит, чтобы точно вычислить целую часть значения R . Возможные решения данной проблемы:

- реализация вещественной длинной арифметики
- сведение решения к целочисленной длинной арифметике
- использование встроенной целочисленной длинной арифметики в языках Python и Java
- использование 16-байтового вещественного типа `__float128` в C++ или типа `Decimal` в Python

Получение значения числа π с высокой точностью так же может быть сложной задачей в условиях олимпиады. На память большинство помнит лишь 2-5 цифр числа, чего явно недостаточно. Использование констант `pi` (паскаль и питон) и `M_PI` (C++), а также вычисление через тригонометрические формулы не позволяют получить значение необходимой точности. Вычисление этого значения "в лоб" с использованием разложения арктангенса в ряд Тейлора с применением длинной арифметики является возможным, но очень трудозатратным выходом. Наиболее простой выход - это использовать встроенный в Windows калькулятор, у которого имеется число π с достаточной точностью в режиме «инженерный».



Задача А. Окружность

```
//Pascal - 90 баллов
var l : extended;

begin
  read(l);
  write(4*(2*trunc(l/2/pi)+1))
end.
```

Контрпример:

99999999999999993624

Правильный ответ:

1273239544735154564

Ошибочный ответ:

1273239544735154572

```
//C++ - 90 баллов
#include <iostream>
#include <cmath>

using namespace std;

int main(){
  long double l;
  long long r;

  cin >> l;
  r = l/8/atan1(1.);
  cout << 4*(2*r+1);

  return 0;
}
```

Задача А. Окружность

```
#Python
#Полное решение №1
from decimal import *
pi = Decimal('3.141592653589793238462643383279502884')
r = int(int(input()) / (2*pi))
print(4*(2*r+1))
```

```
#Python
#Полное решение №2
n = int(input()) * 10**54
pi2 = 2 * 3141592653589793238462643383279502884
r = n // pi2 // 10**18
print(4*(2*r + 1))
```

Задача В. Максимальное число

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Заданы два целых неотрицательных числа A и B .

Требуется найти такое максимально возможное целое число C , которое можно составить как из цифр числа A , так и из цифр числа B .

Входные данные

Входной файл INPUT.TXT содержит два целых неотрицательных числа A и B по одному в каждой строке. Каждое из чисел состоит не более чем из 10^5 цифр.

Выходные данные

В выходной файл OUTPUT.TXT выведите целое число C – ответ на задачу. Если такого числа не существует, выведите «No solution» (без кавычек).

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	240134 794041	4410
2	1234 5678	No solution

Система оценки

Решения, правильно работающие только для чисел, состоящих не более чем из 6 цифр, будут оцениваться в 20 баллов.

Решения, правильно работающие только для чисел, состоящих не более чем из 9 цифр, будут оцениваться в 40 баллов.

Решения, правильно работающие только для чисел, состоящих не более чем из 1000 цифр, будут оцениваться в 60 баллов.

Задача В. Максимальное число

Первоначально считаем числа в строковые переменные SA и SB. Используем два массива счетчиков цифр A и B и определим количество вхождений каждой цифры для каждого из чисел (как в сортировке подсчетом). Понятно, что если цифра D входит в первое число A[D] раз, а во второе - B[D] раз, то в искомое число она войдет $\min(A[D], B[D])$ раз. Для того, чтобы итоговое число было максимальным, следует выводить все цифры в порядке невозрастания, не забывая о том, что возможным ответом может быть 0, который следует вывести только один раз.

Алгоритмическая запись вышеописанного:

```
read(sa, sb) // считываем числа

for i=1..len(sa) a[sa[i]]++ // подсчет вхождений каждой
for i=1..len(sb) b[sb[i]]++ // из цифр в обоих числах

flag = true // флаг: есть ли общие цифры?
for c='9'..'0' // вывод цифр по убыванию
    while(a[c]>0 and b[c]>0) { // вывод цифры с min(a[c], b[c]) раз
        a[c]--; b[c]--
        write(c)
        if(c='0' and flag) exit // если первой цифрой вывели 0...
        flag = false
    }

if(flag) write('No solution') // если нет общих цифр...
```

Задача В. Максимальное число

```
//Pascal
var i : integer;
    c : char;
    flag : boolean;
    sa, sb : string;
    a, b : array['0'..'9'] of integer;
begin
    readln(sa);
    readln(sb);

    flag := true;
    fillchar(a, sizeof(a), 0);
    fillchar(b, sizeof(b), 0);

    for i:=1 to length(sa) do inc(a[sa[i]]);
    for i:=1 to length(sb) do inc(b[sb[i]]);

    for c:='9' downto '0' do
        while (a[c]>0) and (b[c]>0) do begin
            write(c);
            if (c='0') and flag then exit;
            dec(a[c]); dec(b[c]);
            flag := false
        end;

    if flag then write('No solution')
end.
```


Задача С. Киберспорт

(Время: 1 сек. Память: 16 Мб Баллы: 100)

В киберспорте используется множество систем проведения соревнований. Не так давно для игры StarCraft II была введена новая система отбора трех человек из шести на групповой стадии.

Система напоминает круговую: каждый играет с каждым две игры. При этом в такой дуэли один из матчей проходит для каждого игрока на своей выбранной карте (домашняя игра), а другой – на карте, выбранной соперником (гостевая игра). Всего получается 30 игр в турнире. В результате более высокую позицию занимает тот, кто одержит больше побед, если число побед совпадает, то преимущество получает тот, у кого больше побед на гостевых картах, если же и эти параметры равны, то учитывается показатель личных встреч. Если согласно этим правилам в результате невозможно определить первую тройку победителей, то проводятся переигровки.

Требуется написать программу, которая по результатам 30 игр определяет тройку победителей, либо определяет, что нужны дополнительные игры.

Входные данные

Входной файл INPUT.TXT содержит турнирную таблицу 6 x 6. В *i*-й строке сначала идут 6 цифр «0» или «1» (без пробелов) – информация об играх *i*-го игрока на домашних картах, далее через пробел – имя (ник) игрока. На главной диагонали таблицы стоят нули. Если в *i*-й строке и *j*-м столбце стоит «1», то это означает, что *i*-й игрок одержал победу над *j*-м на домашней карте, «0» обозначает проигрыш. Все имена – строки, состоящие не более чем из 10 латинских символов.

Выходные данные

В выходной файл OUTPUT.TXT выведите тройку победителей в алфавитном порядке или слово «Undefined» (без кавычек), если победителей определить невозможно.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	010011 Maru 100010 Solar 000001 Curious 111001 Zest 001001 TY 011100 Sos	Maru TY Zest
2	000010 Snute 001010 Happy 000101 ByuL 110000 Polt 010001 Classic 001100 Lilbow	Undefined

Система оценки

Решения, учитывающие только число выигранных игр, будут оцениваться в 40 баллов.

Решения, определяющие победителей только по числу выигранных игр и по числу выигранных игр на гостевых картах, будут оцениваться в 60 баллов.

Задача С. Киберспорт

Это задача на сортировку структур. Здесь нам нужно упорядочить игроков по трем критериям, описанным в условии задачи. Используем в программе следующие типы данных и функции:

```
//структура "игрок"
struct player{
    int k          //исходный номер игрока
    int score     //общее число побед
    int away      //число гостевых побед
    string nick  //имя игрока
}

//массив игроков
player p[1..6]

//таблица результатов
int a[1..6][1..6]

//функция сравнения игроков по играм
//возвращает истину, если игрок p1
//строго выше игрока p2 по рейтингу
bool Cmp1(player p1, p2)

//функция сравнения игроков по именам
bool Cmp2(player p1, p2)
```

Алгоритм решения задачи:

```
//чтение данных
for i=1..n{
    p[i].k = i
    read(a[i][1..6], p[i].nick)
    for j=1..n{
        p[i].score += a[i][j]
        if(i<>j and a[i][j]==0)
            p[j].score++, p[j].away++
    }
}

//сортировка всех игроков по рейтингу
sort(p[1..6], Cmp1)

//проверка однозначности Топ-3
for i=1..3
    for j=4..6
        if(not Cmp1(p[i],p[j])){
            write('Undefined')
            exit
        }
}

//сортировка Топ-3 по именам
sort(p[1..3], Cmp2)

//вывод Топ-3 игроков по алфавиту
for i=1..3
    writeln(p[i].nick)
```

Задача С. Киберспорт

```
//Pascal
const n=6;

type player = record
    k,score,away : integer;
    nick : string
end;
CmpFunc = function(p1,p2 : player) : boolean;

var i,j : integer;
    s : string;
    a : array[1..n] of string;
    p : array[1..n] of player;

function cmp1(p1,p2 : player) : boolean;
begin
    result := true;
    if p1.score > p2.score then exit;
    if p1.score = p2.score then begin
        if (p1.away > p2.away) or
            (p1.away = p2.away) and
            (a[p1.k,p2.k] > a[p2.k,p1.k]) then exit
        end;
    result := false
end;
end;
```

Задача С. Киберспорт

```
function cmp2(p1,p2 : player) : boolean;
begin
    result := p1.nick < p2.nick
end;

procedure sort(n : integer; cmp : CmpFunc);
var i,j : integer;
    tmp : player;
begin
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if cmp(p[j], p[i]) then begin
                tmp:=p[i]; p[i]:=p[j]; p[j]:=tmp
            end
        end
    end;

begin
    fillchar(p, sizeof(p), 0);
```

```
    for i:=1 to n do begin
        p[i].k := i;
        readln(s);
        a[i] := copy(s,1,pos(' ',s)-1);
        delete(s,1,pos(' ',s));
        p[i].nick := s;
        for j:=1 to n do begin
            inc(p[i].score, ord(a[i,j])-48);
            if (i<>j) and (a[i,j]='0') then begin
                inc(p[j].score); inc(p[j].away)
            end
        end
    end;

    sort(n, cmp1);

    for i:=n div 2+1 to n do
        for j:=1 to n div 2 do
            if not cmp1(p[j],p[i]) then begin
                write('Undefined');
                exit
            end;
        end;

    sort(n div 2, cmp2);

    for i:=1 to n div 2 do
        writeln(p[i].nick)
    end.
```

Задача D. Нортландский шифр

(Время: 1 сек. Память: 16 Мб Баллы: 100)

По результатам исследования одного английского университета, не имеет значения, в каком порядке распознаются буквы в слове. Говорят, чтобы прева и последняя буквы были на месте, остальные буквы могут следовать в любом беспорядке, все равно текст читается без проблем. Причиной этого является то, что мы не читаем каждую букву по отдельности, а все слово целиком.

Тот факт, что Вы смогли прочитать предыдущий абзац, подтверждает его смысл. Однако заметим, что в некоторых словах приведенного примера не только первая и последняя буквы стоят на своих местах, что вероятно повышает шансы на быстрое чтение текста.

Требуется написать программу, которая будет изменять текст, меняя буквы в словах (кроме первой и последней) так, что количество букв, которые останутся на своих позициях, будет минимальным.

Входные данные

Входной файл INPUT.TXT содержит строку, состоящую из слов, разделенных пробелом. Слова состоят из строчных и прописных букв английского алфавита. Максимальная длина строки – 10^5 символов.

Выходные данные

В выходной файл OUTPUT.TXT выведите зашифрованный текст в формате, схожем с форматом входных данных.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	After a storm comes a calm	Atefr a srtom cmeos a clam
2	Adversity is a great teacher	Avtdirsey is a geart taehecr
3	All asses wag their ears	All ases wag teihr eras

Система оценки

Решения, работающие для текста, в котором каждое слово состоит из различных букв, будут оцениваться в 30 баллов.

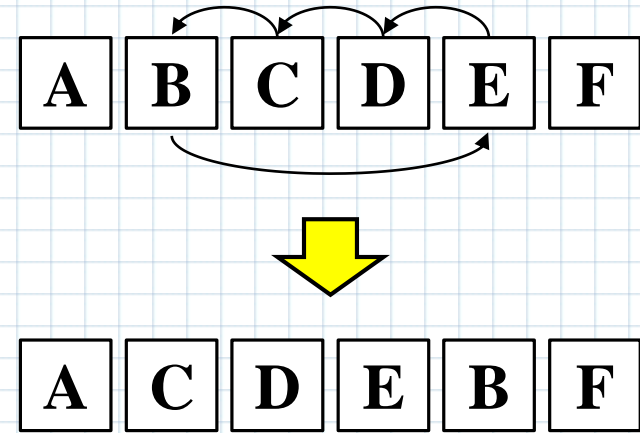
Решения, предполагающие, что во входных данных все слова состоят менее чем из 10 букв, будут оцениваться в 50 баллов.

Задача D. Нортландский шифр

Решение №1 - 30 баллов

Если все буквы в словах различны, то всегда можно получить из строки S длины $N > 3$ строку, где все возможные $N-2$ символа стоят не на своих местах. Действительно, сдвиг подстроки $S[2..N-1]$ на символ влево или вправо решит эту задачу. Строки с меньшей длиной, очевидно, следует оставить без изменений. На языке Python эта задача решается в одну строчку:

```
print(*[s[0]+s[2:-1]+s[1]+s[-1] if len(s)>3 else s for s in input().split()])
```



Решение №2 - 50 баллов

Рассмотрим случай, когда все слова имеют длину, меньшую 10. Это означает, что внутри слова перестановке может быть подвергнуто не более, чем 7 символов. Можно перебрать все возможные перестановки символов для каждого слова. Для одного слова таких вариантов окажется не более 5040 ($=7!$, если все символы различны). Максимальное количество таких слов может быть равно 10000, т.е. худший вариант - перебор 50 млн. значений. Для каждого слова будем выбирать наилучшую перестановку (в которой больше измененных букв) и выводить ее. При этом, если какая то перестановка изменяет положение всех букв, то процесс поиска можно прекратить. Когда много различных букв, то подобная ситуация будет происходить часто. И, наоборот, когда много одинаковых символов, то возможных перестановок значительно меньше. Поэтому перебор получится значительно меньшим, чем верхняя оценка выше.

Задача D. Нортландский шифр

Решение №3 - 100 баллов

Рассмотрим алгоритм перемешивания символов некоторой строки S длины N требуемым образом.

Составим некоторый массив $A[1..N]$ из индексов строки S так, что сначала в нем будут идти все позиции символов "А", потом позиции символов "В" и т.д. до символа "z". Сделать это можно за $O(N)$ методом, схожим на поразрядную сортировку: сначала создадим массив счетчиков букв cnt , далее определим массив начал вхождения ind последовательности каждой из букв, который в дальнейшем используем для заполнения массива A . Заметим, что для заполнения массива A можно использовать и обычную сортировку пар <буква, номер>, но это потребует $O(N \log N)$ действий.

Пусть max - максимальное значение элемента массива cnt (самая высокая частота встречаемости какой-либо буквы). Тогда наилучшим решением задачи будет замена в строке S каждого $A[(k+max-1) \bmod N+1]$ -го символа на символ $S[A[k]]$ прежней строки (сдвиг на max позиций вправо). Для удобства лучше использовать дополнительную строку P :

$$P[A[(k+max-1) \bmod N+1]] = S[A[k]]$$

Общая асимптотика алгоритма - $O(N)$.

S:

A	C	B	A	A	D	C	B	C	A
1	2	3	4	5	6	7	8	9	10

cnt:

4	2	3	1
A	B	C	D

max = 4

ind:

1	5	7	10
A	B	C	D

$ind[A] = 1$
 $ind[i] = ind[i-1] + cnt[i-1]$

A:

1	4	5	10	3	8	2	7	9	6
1	2	3	4	5	6	7	8	9	10
A				B		C		D	

Diagram showing arrows from $ind[A]$ to index 1, $ind[B]$ to index 5, $ind[C]$ to index 7, and $ind[D]$ to index 10. Purple arcs labeled "max" connect index 1 to 5, and index 7 to 10.

P:

C	A	A	C	C	B	A	A	B	D
1	2	3	4	5	6	7	8	9	10

индексы элементов строки S

2	5	1	7	9	8	10	4	3	6
---	---	---	---	---	---	----	---	---	---

Задача D. Нортландский шифр

```
//Pascal - 100 баллов
```

```
var c : char;  
    i, n, m : integer;  
    s, p : string;  
    a : array[0..100005] of integer;  
    cnt, ind : array['A'..'z'] of integer;
```

```
begin
```

```
    while not eof(input) do begin
```

```
        s := '';
```

```
        repeat
```

```
            read(c);
```

```
            if c>' ' then s := s+c;
```

```
        until (c=' ') or eof(input);
```

```
        fillchar(cnt, sizeof(cnt), 0);
```

```
        fillchar(ind, sizeof(ind), 0);
```

```
        n := length(s);
```

```
        for i:=2 to n-1 do inc(cnt[s[i]]);
```

```
        m := cnt['A'];
```

```
        for c:='B' to 'z' do begin
```

```
            ind[c] := ind[pred(c)]+cnt[pred(c)];
```

```
            if cnt[c]>m then m := cnt[c]
```

```
        end;
```

```
        for i:=2 to n-1 do begin
```

```
            a[ind[s[i]]] := i;
```

```
            inc(ind[s[i]])
```

```
        end;
```

```
        p := s;
```

```
        for i:=0 to n-3 do
```

```
            p[a[(i+m) mod (n-2)]] := s[a[i]];
```

```
        write(p, ' ')
```

```
    end
```

```
end.
```


Задача Е. Красивая матрица

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Квадратная матрица – двумерный массив (таблица), составленный из N строк и N столбцов. Строки и столбцы матрицы нумеруются от 1 до N . Число N называют порядком матрицы.

Симметричная матрица – квадратная матрица A порядка N , которая симметрична своему горизонтальному и вертикальному отражению. Более формально: для любой пары (i, j) ($1 \leq i, j \leq N$) верно, что $A_{i,j} = A_{N-i+1,j}$ и $A_{i,j} = A_{i,N-j+1}$.

Красивая матрица – симметричная матрица, составленная из нулей и единиц таким образом, что никакие две клетки, содержащие единицы, не имеют общей стороны.

По заданному значению K требуется составить красивую матрицу, содержащую ровно K единиц. При этом порядок матрицы N должен быть минимально возможным.

Входные данные

Входной файл INPUT.TXT содержит единственное число K ($1 \leq K \leq 10^{19}$) – количество единиц в матрице.

Выходные данные

В первой строке выходного файла OUTPUT.TXT выведите число N – порядок матрицы. Если $K \leq 10^6$, то в последующих N строках выведите N цифр в каждой строке без пробелов – искомую матрицу. Если существует несколько решений, выведите любое.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	4	3 010 101 010
2	9	5 01010 10001 00100 10001 01010
3	1234567	1573

Система оценки

Решения, работающие только для $K \leq 20$, будут оцениваться в 20 баллов.

Решения, работающие только для $K \leq 10^6$, будут оцениваться в 70 баллов.

Задача Е. Красивая матрица

K - количество единиц

N - размерность квадратной матрицы

Несложно понять, что **N** - нечетное число. Действительно, если бы это было не так, то две центральные строки матрицы по горизонтали, так же как и два столбца по вертикали обязаны быть заполнены нулями. Удаление одной центральной строки и одного центрального столбца приведет к сохранению симметрии и значения **K**, но при этом **N** уменьшится на единицу.

Для заданного нечетного **N** максимально возможное количество единиц составит $K = (N^2+1)/2$, т.к. у любой единицы все соседние клетки обязаны быть нулями. Такое максимальное заполнение единицами напоминает шахматную доску, где белые клетки - нули, а черные - единицы.

Покажем, что при $N \geq 5$ всегда возможно разместить любое количество единиц от 1 до K_{\max} . Заполним матрицу единицами в шахматном порядке. Если единица находится не на центральной вертикали или горизонтали, то ей в силу симметрии соответствуют еще другие 3 клетки. Если все такие клетки удалить, то число единиц сократится на 4, а матрица останется красивой. Аналогично, можно удалять симметричные пары клеток на центральной горизонтали или центральной вертикали, при этом количество единиц уменьшится на 2. И наконец, удаление центральной клетки уменьшает число единиц на 1. Такими действиями мы можем уменьшить число единиц с K_{\max} до любого неотрицательного **K**. Следует не забывать о случаях, когда $N < 5$. А именно, это случаи при $K=2$ и $K=3$. их следует рассмотреть отдельно.

По заданному **K** значение **N** можно найти бинарным поиском. Попытка вычисления $\text{ceil}(\sqrt{2*K-1})$ может привести к вычислительной погрешности в силу установленных ограничений на значение **K** в задаче.

1	0	1	0	0	1	0	1
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	1
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1

N - четно



1	0	1	0	1	0	1
0	1	0	0	0	1	0
1	0	1	0	1	0	1
0	0	0	0	0	0	0
1	0	1	0	1	0	1
0	1	0	0	0	1	0
1	0	1	0	1	0	1

N - нечетно

1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1

$$K_{\max} = \frac{N^2 + 1}{2}$$

Задача E. Красивая матрица

```
//Pascal
var i,j,t,n,k : integer;
    m,l,r : int64;
    kx : extended;
    a : array[1..1500, 1..1500] of byte;

begin
  read(kx);

  l := 1; r := 4472135955;
  while l < r do begin
    m := (l+r) div 2;
    if not odd(m) then dec(m);
    if (m-1)/2*(m+1)+1 < kx then l := m+2
      else r := m;
  end;

  if kx=3 then r := 5;
  writeln(r);
  n := r;
  t := (n+1) div 2;

  if kx=2 then begin
    write('000'#13'101'#13'000');
    exit
  end;
  if kx > 1000000 then exit;
```

Задача E. Красивая матрица

```
for i:=1 to n do
  for j:=1 to n do
    a[i,j] := (i+j+1) mod 2;

k := round(n*n div 2 + 1 - kx);

for i:=1 to t-1 do
  for j:=1 to t-1 do
    if (a[i,j]=1) and (k>3) then begin
      a[i,j] := 0; a[i,n-j+1] := 0;
      a[n-i+1,j] := 0; a[n-i+1,n-j+1] := 0;
      dec(k, 4)
    end;
  if k>1 then begin
    a[t-2,t] := 0; a[t+2,t] := 0
  end;
  if odd(k) then a[t,t] := 0;

for i:=1 to n do begin
  for j:=1 to n do write(a[i,j]);
  writeln
end
end.
```