Разбор задач

39 Районная олимпиада школьников Красноярского края по информатике, 7-8 классы (Муниципальный этап ВсОШ по информатике)

8 декабря 2025 г.

ID	Задача	Тема	%
2211	A. Раскраска клеток	Задачи для начинающих	5
2218	B. Шары и коробки	Простая математика	13
2216	C. Будильник	Простая математика	18
2219	D. Робот	Двумерные массивы / Моделирование	31
2220	E. Ребус	Перебор	34

Разбор: Беляев Сергей Николаевич

Задача A. Раскраска клеток

(Время: 1 сек. Память: 32 Мб Баллы: 100)

Ваня нашёл клетчатый листок бумаги размером N×M клеток и нарисовал на нём рамку толщиной в K клеток от границы.

Сколько клеток раскрасил Ваня?

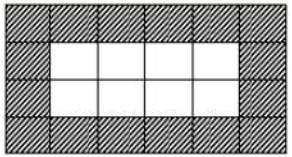
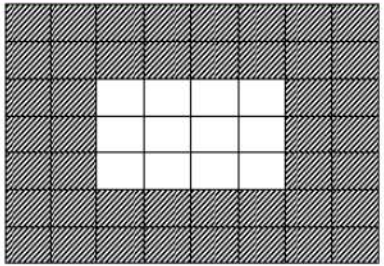
Входные данные

Входной файл INPUT.TXT содержит в трёх строках три целых числа N, M и K по одному на строке соответственно ($1 \le N, M, K \le 10^9$). Гарантируется, что при данных значениях рамку толщиной в K клеток возможно нарисовать.

Выходные данные

В выходной файл OUTPUT.TXT выведите целое число – ответ на задачу.

Примеры

№	INPUT.TXT	OUTPUT.TXT	Пояснение
1	4 6 1	16	
2	7 8 2	44	

Система оценки

Решения, работающие только для K = 1, будут оцениваться в 50 баллов.

Задача A. Раскраска клеток

Дано:

N – высота листка

M – ширина листка

K – толщина рамки

Требуется найти:

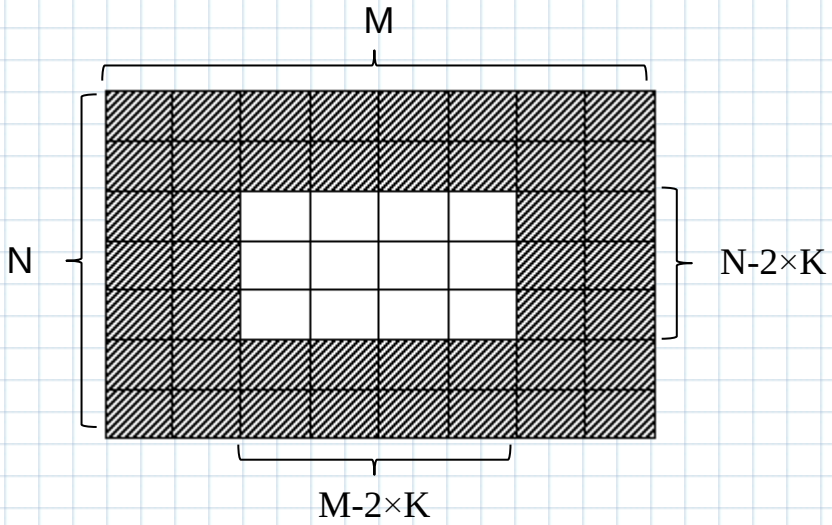
R – количество раскрашенных клеток

Решение:

A = N×M – количество всех клеток

S = (N-2×K) ×(M-2×K) – количество нераскрашенных клеток

R = A – S – ответ на задачу




```
// Алгоритмическая реализация
read(N, M, K)
write(N*M - (N-2*K)*(M-2*K))


#Python
n = int(input())
m = int(input())
k = int(input())
print(n*m - (n-2*k)*(m-2*k))


//C++
#include <bits/stdc++.h>
using namespace std;
long long n, m, k;
int main(){
  cin >> n >> m >> k;
  cout << n*m - (n-2*k)*(m-2*k);
  return 0;
}
```

Задача B. Шары и коробки

(Время: 1 сек. Память: 32 Мб Баллы: 100)

У вас есть A красных и B синих шаров, которые требуется разместить по коробкам. Количество коробок не ограничено. В каждую из используемых коробок можно положить только три шара, при этом шары в одной коробке не могут быть одного цвета. То есть в коробку можно класть либо два красных шара и один синий, либо один красный и два синих.

Требуется определить, можно ли разложить шары по описанным выше правилам по коробкам. Если это возможно сделать, то следует вывести способ такого размещения.

Входные данные

Первые две строки входного файла INPUT.TXT содержит два целых числа A и B – число красных и синих шаров соответственно ($0 \le A, B \le 10^9$).

Выходные данные

В выходной файл OUTPUT.TXT выведите два целых числа через пробел: количество коробок с двумя красными и одним синим шарами и количество коробок с одним красным и двумя синими шарами соответственно.

Если разместить все шары по коробкам, описанным в условии задачи способом невозможно, следует вывести число -1.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	5 4	2 1
2	3 5	-1

Система оценки

Решения, работающие только для A, B ≤ 100, будут оцениваться в 60 баллов.

Задача B. Шары и коробки

Дано:

A – число красных шаров

B – число синих шаров

Требуется найти:

X – количество коробок с 2 красными и 1 синим шаром

Y – количество коробок с 1 красным и 2 синими шарами

Решение:

Поскольку в каждой коробке лежат 3 шара, то A+B должно делится на 3, иначе решения не существует. Мы легко можем вычислить общее число коробок:

$N = (A+B)/3$

Заметим, что в каждой коробке лежит красный шар, а также в X коробках лежит еще по дополнительному красному шару, поэтому $X = A - N$, при этом $A \geq N$, иначе задача не имеет решения.

Аналогично получаем, что $Y = B - N$ и $B \geq N$. Из чего получаем следующую алгоритмическую реализацию:

```
read(A, B)
N = (A+B)//3
if (A+B)%3 > 0 or min(A, B) < N
  write(-1)
else
  write(A-N, B-N)
```

```
#Python
a = int(input())
b = int(input())

n = (a+b)//3

if (a+b)%3 > 0 or min(a, b) < n:
    print(-1)
else:
    print(a-n, b-n)
```

```
//C++
#include <bits/stdc++.h>

using namespace std;

int a, b, n;

int main(){
  cin >> a >> b;
  n = (a+b)/3;
  if((a+b)%3 || min(a,b)<n)
    cout << -1;
  else
    cout << a-n << " " << b-n;
  return 0;
}
```

Задача C. Будильник

(Время: 1 сек. Память: 32 Мб Баллы: 100)

У Пети есть электронные часы с будильником, которые спешат на S минут в час. Однажды вечером перед сном в H_1 часов и M_1 минут он установил на своих часах будильник на утро следующего дня, указав время звонка H_2 часов и M_2 минут.

В какое время прозвенит будильник?

Входные данные

Первая строка входного файла INPUT.TXT содержит время установки будильника (H_1 часов и M_1 минут) в формате ЧЧ:ММ.

Во второй строке в аналогичном формате указано время (H_2 часов и M_2 минут) установленного на часах Пети будильника.

Третья строка входных данных содержит целое число S – количество минут, на сколько каждый час спешат часы у Пети.

Ограничения: $19 \le H_1 \le 23$; $7 \le H_2 \le 9$; $0 \le M_1, M_2, S \le 59$.

Выходные данные

В выходной файл OUTPUT.TXT выведите время, в которое прозвенит будильник в формате ЧЧ:ММ. Гарантируется, что ответ выражается в целых числах.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	19:00 07:00 0	07:00
2	19:00 07:00 4	06:15

Система оценки

Решения, работающие верно только для S = 0, будут оцениваться в 10 баллов.

Задача C. Будильник

Дано:

H_1:M_1 – время установки будильника

H_2:M_2 – время звонка на будильнике

S – количество минут, на которое спешит будильник каждый час

Требуется найти:

H:M – реальное время звонка будильника

Решение:

X – время в минутах от времени установки будильника, через которое будильник прозвонит.

T_1 = 60*H_1 + M_1 – время в минутах установки будильника относительно дня его установки

T_2 = 60*H_2 + M_2 + 1440 – время в минутах установки будильника относительно дня его установки

Через X минут будильник будет спешить на S*X/60 минут, т.е. покажет время, большее стартового на X + S*X/60 минут. С другой стороны, это время равно $T_2 - T_1$. Откуда можно составить уравнение и выразить X:

$$X + S\frac{X}{60} = T_2 - T_1$$

$$60X + SX = 60(T_2 - T_1)$$

$$X(60 + S) = 60(T_2 - T_1)$$

$$X = \frac{60(T_2 - T_1)}{60 + S}$$

```
//Алгоритмическая реализация:
read(h1, m1, h2, m2, s)

t1 = 60*h1 + m1
t2 = 60*h2 + m2 + 1440
x = 60*(t2-t1)//(60+s)
res = t1 + x - 1440

write(res//60, ':', res%60)
```

Задача C. Будильник

```python
#Python
h1, m1 = map(int, input().split(':'))
h2, m2 = map(int, input().split(':'))
s = int(input())
t1 = 60*h1 + m1
t2 = 60*h2 + m2 + 1440
x = 60*(t2-t1)//(60+s)
res = t1 + x - 1440
print('%02d:%02d' % (res // 60, res % 60))
```

```cpp
//C++
#include <bits/stdc++.h>

using namespace std;

int h1, m1, h2, m2, s, res;

int main(){
  scanf("%d:%d\n%d:%d\n%d", &h1, &m1, &h2, &m2, &s);
  res = h1*60 + m1 + 60*((h2+24-h1)*60 + m2-m1)/(60+s) - 1440;
  printf("%02d:%02d", res/60, res%60);
  return 0;
}
```

Задача D. Робот

(Время: 1 сек. Память: 64 Мб Баллы: 100)

Робот рисует спираль, перемещаясь по клетчатому полю. Первоначально он находится в некоторой клетке и закрашивает её. Робот направлен вдоль положительного направления оси OX.

Он действует по следующему алгоритму: совершает D перемещений вперед, затем поворачивает направо и снова делает D перемещений вперед, каждый раз закрашивая клетки, в которых он побывал. После этого робот поворачивает направо и умножает значение D на K. Затем он повторяет описанный процесс. Робот останавливается, сделав суммарно ровно N перемещений.

Требуется вывести карту, на которой отмечены клетки, закрашенные роботом.

Входные данные

В первых трёх строках входного файла INPUT.TXT содержатся целые числа N, D и K по одному на строке ($1 \le N \le 1000$, $1 \le D \le 100$, $2 \le K \le 5$).

Выходные данные

В первой строке выходного файла OUTPUT.TXT выведите два целых числа H и W – высоту и ширину карты. В следующих H строках выведите по W символов. Клетки, посещенные роботом должны содержать символ «#» (ASCII 35), а не посещенные – «.» (ASCII 46). При этом выводимая карта должна иметь минимальную площадь и содержать все посещенные роботом клетки.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	14 2 2	5 5 ###.. #.... #.### #...# #####

Задача D. Робот

```python
#Python
n = int(input())
d = int(input())
k = int(input())

mn = 2005
a = [['.']*mn for _ in range(mn)]
x = y = x1 = y1 = x2 = y2 = mn//2
a[y][x] = '#'
dx, dy = 1, 0

while n>0:
    for _ in range(2):
        for _ in range(d):
            if n>0:
                x += dx
                y += dy
                a[y][x] = '#'
                x1, y1 = min(x1, x), min(y1, y)
                x2, y2 = max(x2, x), max(y2, y)
                n -= 1
        dx, dy = -dy, dx
    d *= k

print(y2-y1+1, x2-x1+1)

for y in range(y1, y2+1):
    print(''.join(a[y][x1:x2+1]))
```

```cpp
//C++
#include <bits/stdc++.h>
#define mn 3000

using namespace std;

int n, d, k, x, y, i, j, dx, dy, x1, y1, x2, y2, a[mn][mn];

void draw(){
  if(n == 0) return;
  a[y+=dy][x+=dx] = 1;
  x1 = min(x1, x);
  y1 = min(y1, y);
  x2 = max(x2, x);
  y2 = max(y2, y);
  n--;
}

int main(){
  cin >> n >> d >> k;
  x = y = x1 = y1 = x2 = y2 = mn/2;
  a[y][x] = 1;
  dx = 1; dy = 0;
  while(n){
    for(i=0; i<2; i++){
      for(j=0; j<d; j++) draw();
      tie(dx, dy) = make_pair(-dy, dx);
    }
    d *= k;
  }
  cout << y2-y1+1 << " " << x2-x1+1 << endl;
  for(y=y1; y<=y2; y++, cout << endl)
    for(x=x1; x<=x2; x++)
      cout << (a[y][x]?'#':'.');
  return 0;
}
```

Задача E. Ребус

(Время: 1 сек. Память: 32 Мб Баллы: 100)

Вам дан математический ребус. Для того, чтобы его разгадать, в свободные ячейки требуется вставить целые числа от 1 до 99 так, чтобы выполнялись равенства.

Ребус представляет собой таблицу 5×5, содержит 2 целых числа A и B, расположенных в верхнем левом и нижем правом углах соответственно. Также в 6 ячейках таблицы содержатся математические операции «+», «–», «*» и «:», обозначающие соответственно сложение, вычитание, умножение и деление. В 6 ячейках записаны знаки «=» – равенства.

13	+		=	
–		*		:
	:		=	
=		=		=
	–		=	7

Рис. 1. Пример ребуса

A	1		=	
2		3		4
	5		=	
=		=		=
	6		=	B

Рис. 2. Схема ребуса

По заданным значениям A и B, а также по заданным 6 знакам арифметических операций в ребусе необходимо вычислить количество возможных решений. При этом заполнять 7 свободных клеток ребуса можно только целыми числами от 1 до 99.

Входные данные

Первые две строки входного файла INPUT.TXT содержат целые числа A и B ($1 \le A, B \le 99$). В третьей строке записана последовательность операций «+», «–», «*» и «:» согласно схеме без разделителей (см. рис. 2), где цифрами обозначен порядок и расположение арифметических операций.

Выходные данные

В выходной файл OUTPUT.TXT выведите одно целое число – количество возможных решений ребуса.

Примеры

№	INPUT.TXT	OUTPUT.TXT	Пояснение
1	10 29 ++++++	153	Одно из 153 возможных решений ребуса: 10 + 3 = 13 + + + 10 + 6 = 16 = = = 20 + 9 = 29
2	13 7 +-*::-	1	Единственное решение ребуса: 13 + 1 = 14 – * : 4 : 2 = 2 = = = 9 – 2 = 7

Система оценки

Решения, работающие только для операций «+», будут оцениваться в 20 баллов.

Решения, работающие только для операций «+» и «–», будут оцениваться в 40 баллов.

Замечание

Для решения данной задачи на языке Python рекомендуется использовать интерпретатор PyPy для ускорения работы программы.

Задача E. Ребус

```python
#Python
a = int(input())
b = int(input())
s = input()

mn = 100

def check(x):
    return x>0 and x<mn

def cnt(x, y, op):
    res = 0
    if op == '+': res = x+y
    if op == '-': res = x-y
    if op == '*': res = x*y
    if op == ':' and y and x%y == 0: res = x//y
    return res*check(res)

ans = 0
for c in range(1, mn):
    for e in range(1, mn):
        for f in range(1, mn):
            d = cnt(a, c, s[0])
            h = cnt(a, e, s[1])
            k = cnt(c, f, s[2])
            g = cnt(e, f, s[4])
            if check(d) and check(h) and check(k) and check(g) and \
               b == cnt(d, g, s[3]) and b == cnt(h, k, s[5]): ans += 1
print(ans)
```

A	0	C	=	D
1		2		3
E	4	F	=	G
=		=		=
H	5	K	=	B

```cpp
//C++
#include <bits/stdc++.h>
#define mn 100

using namespace std;

int check(int x){
  return x>0 && x<mn;
}

int cnt(int x, int y, char op){
  int res = 0;
  if(op == '+') res = x+y;
  if(op == '-') res = x-y;
  if(op == '*') res = x*y;
  if(op == ':' && y && x%y == 0) res = x/y;
  return res*check(res);
}

int solve(int a, int b, string &s){
  int c, d, e, f, g, h, k, res = 0;
  for(c = 1; c < mn; c++)
    for(e = 1; e < mn; e++)
      for(f = 1; f < mn; f++){
        d = cnt(a, c, s[0]);
        h = cnt(a, e, s[1]);
        k = cnt(c, f, s[2]);
        g = cnt(e, f, s[4]);
        if(check(d) && check(h) && check(k) && check(g) &&
           b == cnt(d, g, s[3]) && b == cnt(h, k, s[5])) res++;
      }
  return res;
}

int main(){
  int a, b;
  string s;
  cin >> a >> b >> s;
  cout << solve(a, b, s);
  return 0;
}
```