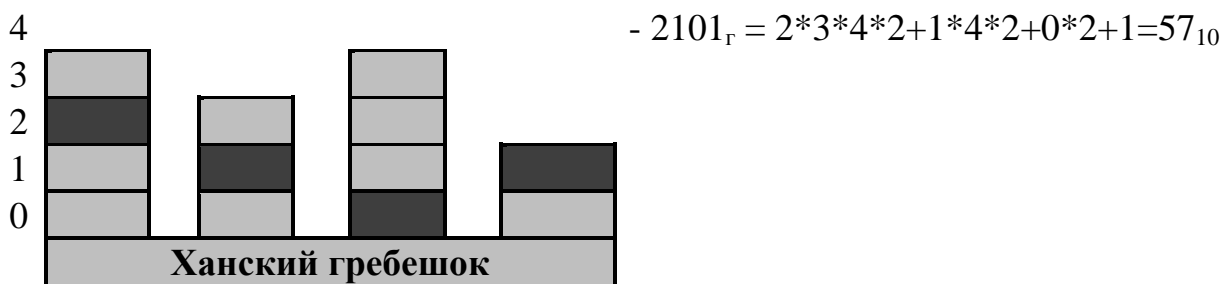


**Межрегиональная предметная олимпиада  
Казанского федерального университета  
по предмету "Информатика"  
Очный тур  
2016-2017 учебный год  
10 класс**

Для программ ограничения по времени 1 с. Ограничения по памяти 256 МБ.

**№1. Гребешковый счет**

В Тридевятом царстве для денежных операций каждому новорожденному выдавались ханские гребешки с разной конфигурацией зубьев. При покупке товаров или получении зарплаты кассовые аппараты меняют золотые метки на зубьях гребешка, отражающих изменение финансового состояния данного жителя Тридевятого царства. С помощью гребешка можно задать число в гребешковой системе счисления. Отличие от обычных в том, что у каждого разряда свое число возможных цифр. У  $i$ -го справа разряда (зубца) цифры от 0 до  $a_i-1$  цифра задается отметкой на зубце.



В Тридевятое царство заморские купцы завезли товары с ценами, представленными в десятичной системе. Помогите определить какие значения на зубьях надо автоматам пометить золотом.

**Входные данные.**

В первой строчке записано число зубьев на гребешке  $N$  ( $N < 10$ ).

Во второй строке записаны высоты зубьев слева направо без пробелов между цифрами, не превышающих 9.

В третьей строке записаны значения золотых меток на зубьях.

В четвертой строке записана цена товара, который житель хочет купить.

**Результат:** Если хватает денег, то выведите, где на гребешке автомат должен пометить золотом зубья слева направо без пробелов между цифрами, иначе "NO" (без кавычек).

Пример	Входные данные	Результат
№1	4 4342 2101 50	0031
№2	4 4342 1201 50	NO

## Решение

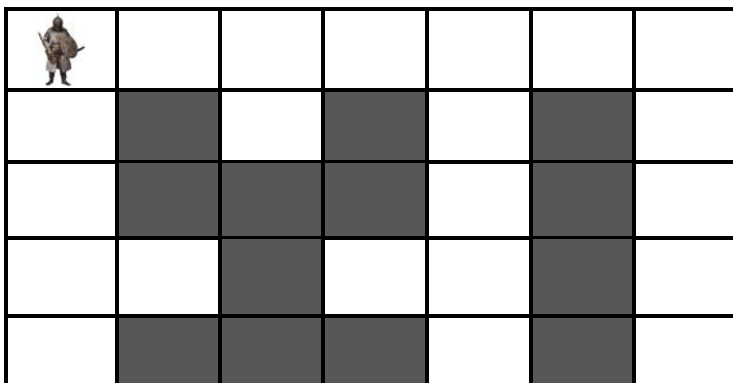
```
var
  n,i,k,s,c:integer;
  a,b:string;
begin
  readln(n); // число зубьев
  readln(a); // высота зубьев
  readln(b); // денежный запас
  read(c); // цена товара
  s:=ord(b[1])-48;
  for i:=2 to n do
    s:=s*(ord(a[i])-48)+ord(b[i])-48;
  if s<c then write(' NO ')
  else
  begin
    s:=s-c;
    b:='';
    for i:=n downto 1 do
      begin
        b:=chr(48+s mod (ord(a[i])-48));
        s:=s div (ord(a[i])-48)
      end;
    write(s)
  end
end.
```

## №2. Робот

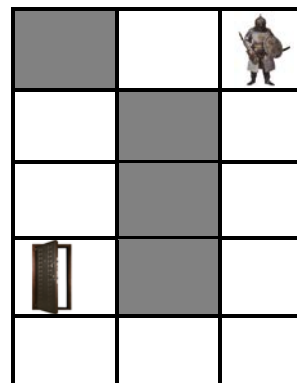
В Тридевятиом царстве разработали робота КФУ-хана, который ходит по лабиринтам. Робот может перемещаться в пустую соседнюю клетку как через сторону, так и через угол по диагонали.

Нужно сосчитать число различных путей выхода из заданного лабиринта, где темным цветом отмечены непроходимые клетки. Робот в пройденную клетку не может входить дважды.

Заданный лабиринт





Пример лаб-та




- 10 различных путей

### Решение

	2		8		32	
2				24		64
2				24		
8				96		
8				96		64
	8		32		128	

$$3 \cdot (32 + 96 + 128) = 768$$

### №3. Судоку

Описание классической игры СУДОКУ. Имеется игровое поле размера 9x9 клеток, в некоторых клетках находятся цифры от 1 до 9, а остальные – пустые. Нужно заполнить пустые клетки цифрами от 1 до 9 так, чтобы в каждой строке, каждом столбце и каждом маленьком квадратике (выделены жирными линиями) встречались все цифры от 1 до 9 ровно по 1 разу.

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

Дано заполненное поле для игры СУДОКУ. Написать функцию (программу), которая проверит правильность его заполнения.

**Входные данные:** 9 строк по 9 цифр от 1 до 9, между цифрами в строке стоит ровно один пробел.

**Результат:** “ДА”, если поле заполнено правильно, или “НЕТ” в противном случае.

### Решение

Написать 3 функции для проверки правильности заполнения

1. Любой строки
2. Любого столбца
3. Любого квадрата размера 3 на 3

Если считать, что матрица Sudoku [9][9], содержащая заполненное поле, является глобальной, то первая функция на языке Си++ может для примера выглядеть так:

```

Bool CheckRow (int row)
{
    bool dig [9]; // в этом массиве будем отмечать
                  // встреченные в строке цифры
    for ( int i=0; i < 9; i++ )
        dig [i] = false;
    for ( int i=0; i < 9; i++ )
        dig [Sudoku [row, i] - 1] = true;
    for ( int i=0; i < 9; i++ )
        if ( ! dig [i] ) return false;
    return true;
}

```

Вторая и третья функции имеют небольшие отличия.

В главной функции нужно ввести исходный массив и организовать вызов этих функций для каждой строки, каждого столбца и для каждого маленького квадрата 3 на 3. Для правильного заполнения поля sudoku все вызовы должны вернуть ИСТИНУ!

#### №4. Формула

В Тридевятом царстве робота КФУ-хана решили использовать при обучении самых способных учеников абстрактным наукам.

Имеются 4 основные операции над неотрицательными целыми числами, используемые в научных расчетах Тридевятого царства:

1.  $X \text{ k } Y = X + Y$
2.  $X \text{ a } Y = Y - X$
3.  $X \text{ t } Y = X * Y$
4.  $X \text{ b } Y = X / Y$  – деление нацело

Сложные научные изыскания производятся с помощью формул, где порядок выполнения определяется скобками. При делении на 0 вычисление надо прервать и сообщить «Деление на 0.» (без кавычек с точкой в конце).

Напишите для робота КФУ-хана программу, определяющую числовое значение заданного выражения.

**Входные данные:** Вводится формула, значения результатов любых операций не превышают  $10^9$ . Длина строки-формулы не превышает  $10^5$ .

**Результат:** выводится результат вычисления или сообщение о том, что происходит деление на 0.

Примеры	Входные данные	Результат
№1	$(((((32k12)a62)k(2t4))b2)$	13
№2	$(((((32k12)a62)k(2t4))b(2a2))$	Деление на 0.

#### Решение

1. Стек операций и стек операндов инициализируются.
2. Пока есть символы вводится символ S.

- 2.1. Если символ  $S = \langle\langle \rangle\rangle$ , то заносится в стек операций.
- 2.2. Если символ  $S$  равен  $\langle\langle * \rangle\rangle$ ,  $\langle\langle + \rangle\rangle$  или  $\langle\langle - \rangle\rangle$ , то
  - 2.2.1. ранее выбранное число  $R$  заносится в стек операндов,
  - 2.2.2. в стек операций заносится  $S$ .
- 2.3. Если  $S$  – цифра, то из текста вводится число  $R$ .
- 2.4. Если  $S = \langle\langle \rangle\rangle$ , то
  - 2.4.1. Берутся два последних элемента  $Op1$  и операнд  $Op2$  из стека операндов.
  - 2.4.2. Выбирается операция  $O$ .
  - 2.4.3. Если  $O = \langle\langle / \rangle\rangle$  и  $O2 = 0$ , то
    - 2.4.3.1. Выдается сообщение «Деление на 0.»
    - 2.4.3.2. Останов!
  - 2.4.4. Выполняется операция  $O$  и записывается в  $R$ .
  - 2.4.5. В стеке операндов последние два операнда удаляются и записывается  $R$ .
  - 2.4.6. Из стека операций удаляются операция  $O$  и  $\langle\langle \langle \rangle\rangle$ .
3. Из стека операций выводится последнее значение

### №5. Незнайка – художник

В Солнечном городе Тридевятого царства Незнайка дружил с Карандашом. И ему нравилось, как он рисует.

Незнайка придумал автомат по рисованию на холсте размерности  $N \times M$ . Этот автомат устанавливался в верхнем левом углу холста с координатой  $(0,0)$ . Затем Незнайка закладывал капсулу с голубой краской и стрелял из него прямоугольниками с координатами  $(X_1, Y_1, X_2, Y_2)$  на холсте. Краска при полете разбрызгивалась во все точки холста, которые оказывались между этими координатами.

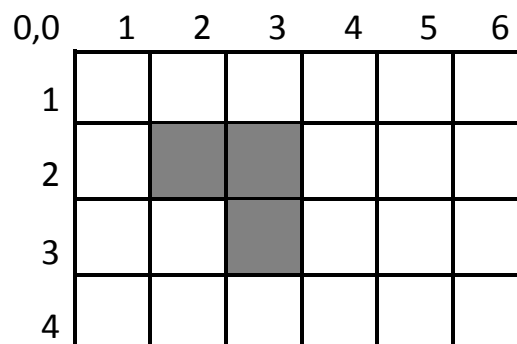
Нужно определить размер окрашенной площади после  $K$  выстрелов из данного автомата.

#### **Входные данные:**

В первой строке вводного файла INPUT.TXT заданы размеры холста  $N, M$  ( $0 < N, M < 1000$ ) и число выстрелов  $K$  ( $K < 1000$ ).

Далее в  $K$  строках заданы координаты выстрелов. Все координаты выстрелов принадлежат холсту.

**Выходные данные:** В выводной файл вывести размер окрашенной площади.



$N=6, M=4, K=2$

$(X_{11}, Y_{11}, X_{21}, Y_{21}) = (1, 1, 3, 2)$

$(X_{21}, Y_{21}, X_{21}, Y_{21}) = (2, 1, 3, 3)$

Примеры	Входные данные	Результат
№1	10 10 2 1 1 3 2 2 1 3 3	4
№2	10 10 1 5 5 10 10	25

## Решение

```
var
  n,m,i,j,k,r,x1,y1,x2,y2:integer;
  a:array[1..1000,1..1000] of byte; //холст
begin
  readln(n,m,k);
  for i:=1 to m do
    for j:=1 to n do
      a[i,j]:=0;
  for r:=1 to k do
    begin
      read(x1,y1,x2,y2);
      for i:=x1+1 to x2 do
        for j:=y1+1 to y2 do
          a[i,j]:=1 // окрас холста
    end;
  r:=0;
  for i:=1 to m do
    for j:=1 to n do
      r:=r+a[i,j]; // подсчет окрашенных точек
  write(r)
end.
```

### №6. Коробка фломастеров

Как-то раз Карандаш подарил Незнайке  $N$  совершенно одинаковых фломастеров и  $M$  коробок для фломастеров, в каждом из которых помещается не более  $K$  фломастеров. Тогда Незнайка задался вопросом, сколькими способами можно разложить все  $N$  фломастеров по коробкам. Помните, что фломастеры все одинаковые, и если вы поменяете два фломастера местами, то это не образует новый способ. Коробки тоже все одинаковы, то есть если полностью обменять содержимое двух коробок, то это будет тот же самый способ. Кроме того, Незнайка согласен на то, чтобы какие-то коробки были пустыми. Помогите ему посчитать количество способов.

#### **Входные данные:**

В первой строке вводного файла INPUT.TXT заданы три целых числа через пробел:  $N, M, K$  ( $0 < N < 2500, 0 < M, K < 50$ )

**Выходные данные:** В выводной файл вывести количество способов.

Примеры	Входные данные	Результат	Пояснения
№1	4 2 3	2	(3,1), (2,2), – первое число в скобках – число фломастеров в одной коробке и второе – в

			другой коробке.
№2	4 2 5	3	(2,2), (3,1), (4,0)
№3	7 2 3	0	Нельзя разложить 7 фломастеров

### Решение

```

import java.io.*;
import java.math.BigInteger;
import java.util.StringTokenizer;

/**
 * Created by user on 27.02.2017.
 */
public class Main10 {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out));
        init(in);
        solve();
        print(out);
        in.close();
        out.close();
    }

    static int n, m, k;
    static BigInteger[][][] dp;
    static BigInteger ans = new BigInteger("0");

    private static void init(BufferedReader in) throws IOException {
        StringTokenizer st = new StringTokenizer(in.readLine());
        n = Integer.parseInt(st.nextToken());
        m = Integer.parseInt(st.nextToken());
        k = Integer.parseInt(st.nextToken());
    }

    private static void solve() {
        dp = new BigInteger[m][n + 1][k + 1];
        k = Math.min(k, n);
        for (int i = 0; i < k + 1; i++)
            dp[0][i][i] = new BigInteger("1");
        for (int j = 1; j < m; j++)

```

```

for (int i = 0; i < Math.min(n,(j+1)*k) + 1; i++)
    for (int w = 0; w < Math.min(i, k) + 1; w++) {
        if (w*(j+1)>n) break;
        dp[j][i][w] = new BigInteger("0");
        for (int q = w; q <= k; q++) {
            if (q*j+w>i) break;
            if (dp[j - 1][i - w][q] != null) {
                dp[j][i][w] = dp[j][i][w].add(dp[j - 1][i - w][q]);
            }
        }
    }
if (dp[m - 1][n] != null) {
    for (int w = 0; w < k + 1; w++){
        if (dp[m - 1][n][w] != null)
            ans = ans.add(dp[m - 1][n][w]);
    }
}
private static void print(PrintWriter out) throws IOException {
    out.print(ans.toString());
}
}

```