

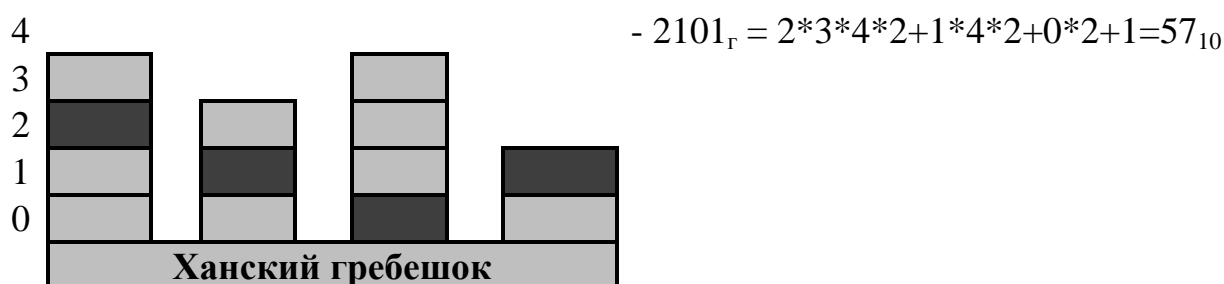
**Межрегиональная предметная олимпиада  
Казанского федерального университета  
по предмету "Информатика"  
Очный тур  
2016-2017 учебный год  
9 класс**

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.

Например:



В Тридевятое царство заморские купцы завезли товары с ценами, представленными в десятичной системе. Помогите жителям этой страны определить, хватает ли денег для покупки товара.

**Входные данные.**

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

Во второй строчке записаны высоты зубьев слева направо без пробелов между цифрами, не превышающие 9.

В третьей строчке записаны значения золотых меток на зубьях.

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

**Результат:** Если хватает, то выводится "YES", иначе "NO".

<b>Примеры</b>	<b>Входные данные</b>	<b>Результат</b>
№1	4 4342 2101 50	YES
№2	4 4342 1201 50	NO

## Решение

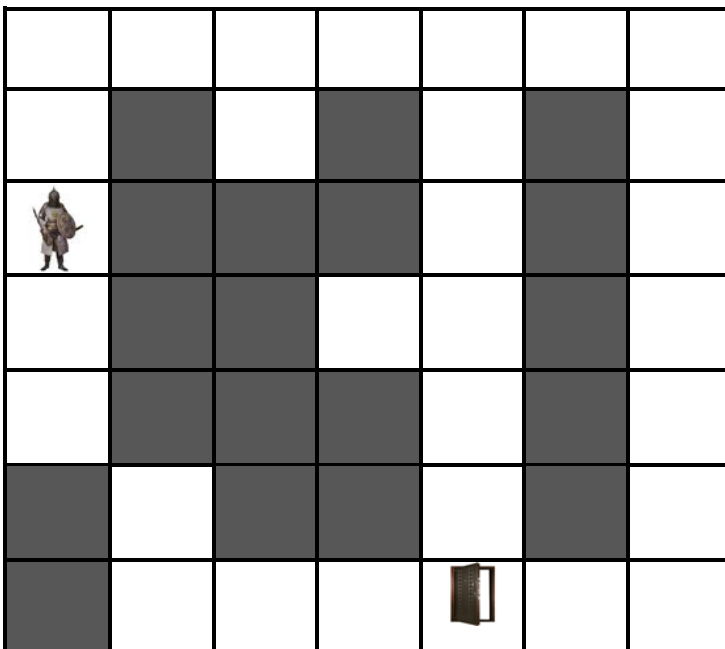
```
var
  n,i,k,s,c:integer;
  a,b:string;
begin
  readln(n); // число зубьев
  readln(a); // высота зубьев
  readln(b); // денежный запас
  read(c); // цена товара
  s:=ord(b[1])-48;
  for i:=2 to n do
    s:=s*(ord(a[i])-48)+ord(b[i])-48;
  if s>=c then write('YES')
  else write('NO')
end.
```

## №2. Робот

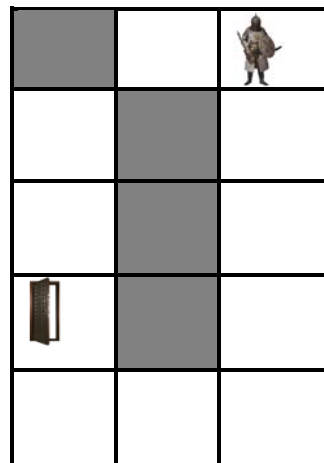
В Тридевятиом царстве разработали робота КФУ-хана, который ходит по лабиринтам. Робот может перемещаться в пустую соседнюю клетку как через сторону, так и через угол по диагонали.

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

Заданный лабиринт





Пример лаб-та



- 10 различных путей

## Решение

1	2	4	8		32	
---	---	---	---	--	----	--

1		4		24		64
				24		
1						
1				96		
	1			96		64
	1	2	2		128	

$$3 \cdot (2 + 96 + 128) = 678$$

### №3. Судоку

Описание классической игры СУДОКУ. Имеется игровое поле размера 9 на 9 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.

Решить приведенный пример – заполнить пустые клетки. (программу писать не требуется, нужен только ответ на конкретный пример)

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

### Решение

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

### №4. Формула

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

Имеются 3 основные операции над целыми неотрицательными числами, используемые в научных расчетах Тридевятого царства:

1.  $X \text{ k } Y = X + Y$
2.  $X \text{ a } Y = Y - X$
3.  $X \text{ t } Y = X * Y$

Сложные научные изыскания производятся с помощью формул, где порядок выполнения определяется скобками.

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

**Входные данные:** в строке записана формула, значения результатов любых операций не превышают  $10^9$ . Длина строки-формулы не превышает  $10^3$ .

**Результат:** число.

Пример	Входные данные	Результат
№1	$((32k12)a62)k(2t4)$	26

### Решение

1. Стек операций и стек операндов инициализируются.
2. Пока есть символы вводится символ S.
  - 2.1. Если символ S = «(», то заносится в стек операций.
  - 2.2. Если символ S равен «\*», «+» или «-», 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 = «)», то
    - 2.4.1. Берутся два последних элемента Op1 и операнд Op2 из стека операндов.
    - 2.4.2. Выполняется операция O, записанная в стеке операций и записывается в R.
    - 2.4.3. В стеке операндов последние два операнда удаляются из стека операций удаляются операция и «(». В стек заносится R.
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 особой XOR краской и стрелял из него в координату (X,Y) на холсте, где отсчет шел от левого верхнего угла. Капсула с краской при полете разбрызгивалась во все точки холста, которые оказывались слева и сверху от траектории полета. XOR краска, попав на холст, высыхала, а при соприкосновении с сухой XOR краской, растворяла ее и испарялась. Нужно определить размер окрашенной площади после K выстрелов из данного автомата.



---

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

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

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

**Входные данные:**

В первой строке вводного файла заданы размеры холста  $N, M$  ( $0 < N, M < 1000$ ) и число выстрелов  $K$  ( $K < 100$ ).

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

**Результат:** В выводной файл выводится размер окрашенной площади.

Примеры	Входные данные	Результат
№1	6 4 1 1 2	2
№2	6 4 2 1 3 3 1	4

**Решение**

```
var
  n,m,i,j,k,r,x,y: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(x,y);
      for i:=1 to x do
        for j:=1 to y do
          a[i,j]:=1-a[i,j] // 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
        end;
      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. А вот коробки все пронумерованы. 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:  $N, M, K$  ( $0 < N < 400, 0 < M, K < 20$ )

**Результат:** В выводной файл выводится количество способов.

Примеры	Входные данные	Результат	Пояснения
№1	4 2 3	3	три способа: (1,3), (2,2), (3,1) – тут в скобках первое число – число фломастеров в первой коробке и второе – во второй коробке.
№2	4 2 5	5	(0,4), (1,3), (2,2), (3,1), (4,0)
№3	7 2 3	0	Незнайка никак не сможет разложить 7 фломастеров

### Решение

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

/**
 * Created by user on 27.02.2017.
 */
public class Main9 {
    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 String ans = "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 = Math.min(k, n);
    for (int i = 0; i < k + 1; i++)
        dp[0][i] = new BigInteger("1");

    for (int j = 1; j < m; j++)
        for (int i = 0; i < n + 1; i++) {
            dp[j][i] = new BigInteger("0");
            for (int q = 0; q < Math.min(i, k) + 1; q++) {
                if (dp[j - 1][i - q] != null)
                    dp[j][i] = dp[j][i].add(dp[j - 1][i - q]);
            }
        }
    if (dp[m - 1][n] != null) {
        ans = dp[m - 1][n].toString();
    }
}

private static void print(PrintWriter out) throws IOException {
    out.print(ans);
}
}

```