

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



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

№1. 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

■		
	■	
	■	
	■	

- 10 различных путей

Решение

 1	2	4	8		32	32	
2	■	4	■	24	■	■	64
2	■	■	■	24	■	128	128
4	4	■	48	48	■	■	256

8				96			
8					96		256
	8	16	32			512	
	8		32				

$$32+32+7*96+6*512=64+6*608=3808$$

№2. Судок

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 квадрате (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

КС	КС	КС
КС	КС	КС
КС	КС	КС

Дано заполнение полей для игры 9 классических СУДОКУ (КС), расположенных по три в строке. Входные данные не противоречивы (цифры от 1 до 9 в каждой строке и столбце). Написать программу, которая проверяет правильность заполнения игровых полей.

Входные данные: 27 строк по 27 цифр, между цифрами в строке ровно один пробел.

Результат: Если эти СУДОКУ правильно заполнено, то вывести “YES”, иначе “NO” (без кавычек).

Решение

Написать 3 функции для проверки правильности заполнения

1. Любой строки
2. Любого столбца
3. Любого квадрата размера 3 на 3
4. Любого классического судок

Если считать, что матрица Sudoku [27][27], содержащая заполненное поле, является глобальной, то первая функция на языке Си++ может для примера выглядеть так:

```

Bool CheckRow (int x, // номер столбца КС
               int y, // номер строки КС
               int row // номер строки в КС)

```

```
{
```

```

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 [y*9+row,x*9+ i] - 1] = true;
for ( int i=0; i < 9; i++ )
    if ( ! dig [i] ) return false;
return true;
}

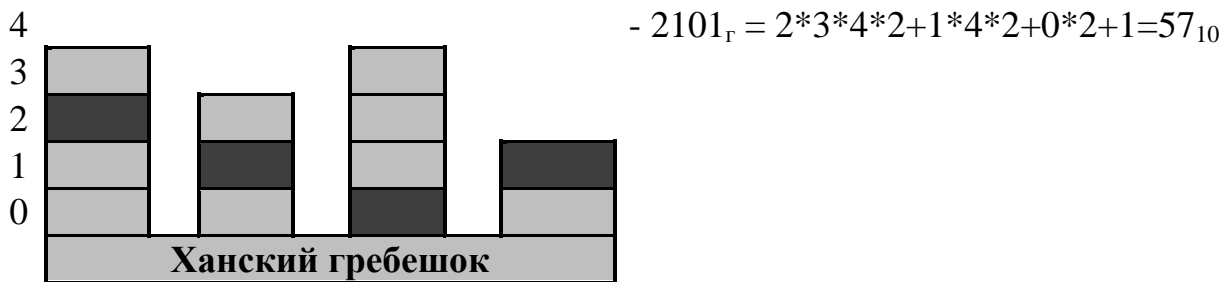
```

Вторая и третья функции имеют небольшие отличия.

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

№3. 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 Тридевятом царстве любимый праздник – «Туй» (свадьба), на котором все гости пляшут, песни поют, молодоженам счастья желают и иногда угощаются. На этом празднике каждый гость стремится помочь молодоженам и передает главе нового семейства половина своего денежного запаса (частное от деления на 2) . Все деньги, которые не могут уместиться на гребешке главы семейства идут на угощение гостей.

Помогите определить, какая сумма в десятичной системе счисления пойдет на угощение гостей.

Входные данные:

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

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

В третьей строке записаны значения золотых меток на зубьях гребешка главы семейства.

В четвертой строке записано число гостей. Далее по 3 строки записаны сведения о наличии денег каждого гостя.

Результат: Выдается сумма оставшаяся для угощения гостей если она есть. Если все переданные деньги ушли на пополнение бюджета молодоженов, то надо вывести "NO"

Пример	Входные данные	Результат
№1	4 4342 2101 2 3 234 121 2 47 34	8

Решение

```
var
  maxc,m,n,i,k,s,c,g,r,cg:integer;
  u,v,a,b:string;
begin
  readln(n); // число зубьев
  readln(a); // высота зубьев
  readln(b); // денежный запас
  // имеющаяся сумма на ханском гребешке
  s:=ord(b[1])-48;
  for i:=2 to n do
    s:=s*(ord(a[i])-48)+ord(b[i])-48;
  // возможная максимальная сумма на гребешке
  maxc:= ord(a[1])-49;
  for i:=2 to n do
    maxc:=maxc*(ord(a[i])-48)+ord(a[i])-47;
  // подсчет подарков гостей
  readln(g); // число гостей
  cg:=0; // сумма подарка гостей
  for r:=1 to g do
    begin
      readln(m); // число зубьев
      readln(u); // высота зубьев
      readln(v); // денежный запас
      s:=ord(b[1])-48;
      for i:=2 to m do
        s:=s*(ord(v[i])-48)+ord(u[i])-48;
```

```

cg:=cg+s div 2
end;
if maxc-s>=cg then write(' NO ')
else // подсчет суммы угощения гостей
write(cg-(max-s))
end.

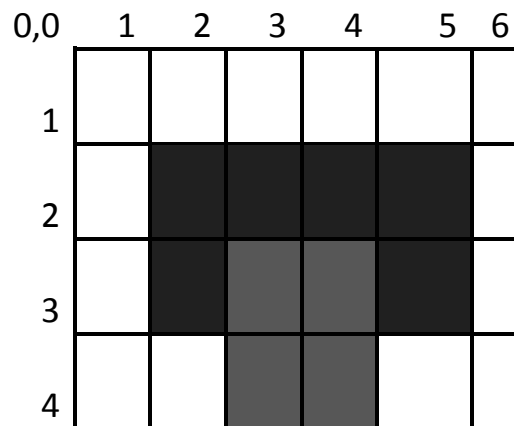
```

№5. Незнайка – художник

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

В солнечном городе продавались холсты белого цвета с маркировкой W и только краски трех цветов: голубой с маркировкой B, зеленый – G, алый – R.

Незнайка придумал автомат по рисованию на холсте размерности NхM. Этот автомат устанавливался в верхнем левом углу холста с координатой (0,0). Затем Незнайка закладывал капсулу с одной из красок C и стрелял из него прямоугольниками с координатами (X₁,Y₁, X₂,Y₂) на холсте. Краска при полете разбрызгивалась во все точки холста, которые оказывались между этими координатами.



N=6, M=4

C=B (X₁₁,Y₁₁,X₂₁,Y₂₁)=(1,1,5,3)

C=R (X₁₁,Y₁₁,X₂₁,Y₂₁)=(2,2,4,4)

Нужно определить размеры окрашенных площадей каждой из красок после K выстрелов из данного автомата.

Входные данные:

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

Далее в K строках заданы цвета красок в капсуле и координаты выстрелов. Все координаты выстрелов принадлежат холсту.

Выходные данные: В выводной файл построчно вывести размеры окрашенных данным цветом площадей, в формате: маркировка (пробел) количество, в порядке: B, G, R, W.

Примеры	Входные данные	Результат
№1	10 10 2 B 1 1 3 5 R 2 2 4 4	B 6 G 0 R 4 W 14
№2	10 10 1 G 5 5 12 13	B 0 G 25 R 0 W 75

Решение

```

var
n,m,i,j,k,r,x1,y1,x2,y2,cb,cg,cr:integer;
c:char;
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(c,x1,y1,x2,y2);
        case c of
          'B':r:=1;
          'G':r:=2;
          'R':r:=4;
        end;
        for i:=x1+1 to x2 do
          for j:=y1+1 to y2 do
            a[i,j]:=r // окрас холста
          end;
        end;
        cb:=0;
        cg:=0;
        cr:=0;
        for i:=1 to m do
          for j:=1 to n do
            case a[i,j] of // подсчет окрашенных точек
              1: cb:=cb+1;
              2: cg:=cg+1;
              4: cr:=cr+1;
            end;
          writeln('B ',cb,);
          writeln('G ',cg,);
          writeln('R ',cr,);
          write('W ',n*m-cb-cg-cr,);
        end.

```

№4. Формула

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

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

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

Приоритет операций:

1. ()
2. b , t

4. $X \text{ b } Y = X / Y$ – деление нацело

3. a, k

Сложные научные изыскания производятся с помощью формул, где порядок выполнения определяется скобками и приоритетом операции. При делении на 0 вычисление 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.

Пример	Входные данные	Результат
№1	32k(12a62)k2t4	90

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. Если в стеке операций находится операция O, равная «+», «-» «*» или «/», то
 - 2.2.1.1. Берутся два последних элемента Op1 и операнд Op2 из стека операндов.
 - 2.2.1.2. Если «O»= «/» и Op2=0,
 - 2.2.1.2.1. Вывод «Деление на 0.»
 - 2.2.1.2.2. Останов!
 - 2.2.1.3. Выполняется операция O, записанная в стеке операций и записывается в R.
 - 2.2.1.4. В стеке операндов последние два операнда удаляются. В стек заносится R.
 - 2.2.1. иначе если в стеке операций находится операция O, равная «+» или «-», то
 - 2.2.1.5. Берутся два последних элемента Op1 и операнд Op2 из стека операндов.
 - 2.2.1.6. Выполняется операция O, записанная в стеке операций и записывается в R.
 - 2.2.1.7. В стеке операндов последние два операнда удаляются. В стек заносится R.
 - 2.2.1. иначе если в стеке операций находится операция O, равная «(», то в стеке операндов заносится S.
 - 2.3. Если S – цифра, то
 - 2.3.1. Из текста выбирается число R.
 - 2.3.2. R заносится в стек операндов.
 - 2.4. Если символ S равен «)», то
 - 2.4.1. Пока в стеке операций последняя операция O != «(»
 - 2.4.1.1. Берутся два последних операнда Op1 и Op2 из стека операндов.
 - 2.4.1.2. Выполняется операция O, записанная в стеке операций и записывается в R.
 - 2.4.1.3. Операция O удаляется из стека операций
 - 2.4.1.4. В стеке операндов последние два операнда удаляются. В стек заносится R.
 - 2.4.2. Из стека операций удаляется символ «(».
3. Из стека операций выводится последнее значение

№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, так чтобы в каждой было не менее R фломастеров. 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. Помогите ему посчитать количество способов.

Входные данные:

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

Результат: В выводной файл вывести количество способов.

Примеры	Входные данные	Результат	Пояснение
№1	4 2 3 1	2	(3,1), (2,2), – первое число в скобках – число фломастеров в одной коробке и второе – в другой коробке.
№2	4 2 5 0	3	(2,2), (3,1), (4,0)
№3	4 2 3 2	1	(2,2)
№4	4 2 5 1	2	(2,2), (3,1)
№5	7 2 3 1	0	Незнайка 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 Main11 {
    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, r;
    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());
    r = Integer.parseInt(st.nextToken());
}
private static void solve() {
    n = n - m*r;
    if (n<0) return;
    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());
}
}

```