#### Задача про нефтяные пятна

бродит по интернету с 2010 года. Вот ее условие:

*После аварии на морской нефтяной скважине в океан вылилось много нефти. Она растеклась по воде, после чего образовалась определенное количество нефтяных пятен. Для ликвидации последствий аварии был создан штаб по координации действий. Сотрудники штаба хранят информацию о пятнах в компьютере в виде матрицы размерностью M x N. Ячейка матрицы содержит 0, если нефтяное пятно в этих координатах отсутствует и 1, если есть (2 ≤ M, N ≤ 100). В матрице ячейки пятен не могут касаться друг к другу ни сторонами, ни углами.*

*Задание*. Для облегчения ликвидации последствий аварии нужно написать программу OIL .\*, которая будет находить общее количество пятен и количество пятен с одинаковой площадью.

*Входные данные*. Входной текстовый файл OIL.DAT содержит в первой строке два числа M и N, далее следуют M строк, в каждой по N целых чисел разделенных пробелами - элементы матрицы.

*Исходные данные*. Выходной текстовый файл OIL.SOL содержит в первой строке целое число k - общее количество пятен, далее в каждом из рядов содержится по два числа, первое - площадь пятна, второе - их количество. Данные сортировать по площадям в порядке возрастания.

Изображения задания, входного и выходного файлов можно увидеть, например, [ЗДЕСЬ](http://www.cyberforum.ru/post1221968.html).

Итак, начинаем с разбора условия.

Нам необходимо считать размер матрицы - двумерного массива, числа m и n. Поскольку известно, что *(2 ≤ M, N ≤ 100),*то наша матрица при объявлении переменных может быть заданна так:

Var

a:array[2..100,2..100] of integer;

Подробнее о матрицах можно почитать в предыдущих уроках - [**Матрица**](http://informatika117pas.blogspot.com/2014/10/7.html)**и**[**Решение задач с матрицей**](http://informatika117pas.blogspot.com/2014/10/8.html)

Так вот, матрица у нас заполнена 0 и 1, а значит, найдя сумму всех элементов матрицы, мы могли узнать общую площадь загрязнения.

ВОТ И НАПИШИТЕ с целью тренировки такую программку. Сами.

А трудность в решении поставленной задачи состоит в том, что необходимо с одной стороны посчитать каждую загрязненную клетку не саму по себе, а вместе с соседними загрязненными, и при этом ни одну из них не посчитать дважды.

Для того, чтоб полностью разобраться с решением задачи, я рекомендую ознакомиться с понятиями "Процедура" в языке Paskal, и "Рекурсия" в математике и программировании.

[**Процедуры и функции в Паскале - 1**](http://pascal.guti.ru/procedure.html)

[**Процедуры и функции в Паскале - 2**](http://informatics.mccme.ru/mod/book/view.php?id=529)

[**Процедуры и функции в Паскале - 3**](http://pascal.org.ua/procedures-functions)

[**Рекурсия в программировании - 1**](http://comp-science.narod.ru/Progr/Rekursia.html)

[**Рекурсия в Паскале - 2**](http://www.pascal.helpov.net/index/recursion_pascal_programming)

[**Рекурсия в Паскале - 3**](http://edunow.su/site/content/pascal/rekursii_v_pascal)

К следующей олимпиаде, которая будет через год, мы уже обязательно разберемся с этими понятиями. А пока можете посмотреть решение базовой задачи:

### Задача про нефть. Решение.

Решение данной задачи любезно предоставлено старшим преподавателем Одесской Национальной Пищевой Академии Попковым Д.Н.

Комментарии отделены //двойным слешем.

program Project1;

var
    f1,f2:textfile;
    m,n,i,j,k,l:integer;
    a,b:array [0..100,0..100] of integer;
    c:array [0..10000] of integer;//хранит площадь пятен под i-м номером

procedure qSort(l,r:longint);  //сортировка
var i,j:longint;
    w,q:integer;
begin
  i := l; j := r;
  q := c[(l+r) div 2];
  repeat
    while (c[i] < q) do inc(i);
    while (q < c[j]) do dec(j);
    if (i <= j) then
    begin
      w:=c[i]; c[i]:=c[j]; c[j]:=w;
      inc(i); dec(j);
    end;
  until (i > j);
  if (l < j) then qSort(l,j);
  if (i < r) then qSort(i,r);
end;

begin
assign(f1,'OIL.DAT');
reset(f1);
assign(f2,'OIL.SOL');
rewrite(f2);
readln(f1, m, n);
k:=0;//хранит номер последнего определённого пятна
for i:=0 to m-1 do
  for j:=0 to n-1 do
    begin
    read(f1, a[i,j]);
    b[i,j]:=0; //массив b хранит 0 если область чиста, и номер пятна, если загрязнена
    if a[i,j]=1 then //если область загрязнена, то проверяем ячейку слева, сверху и слева-сверх по диагонали
      if ((i>0)and(b[i-1,j]<>0)) then  //сверяем с ячейкой слева, загрязнена ли она чем-либо
          begin
          b[i,j]:=b[i-1,j];//если да, то ячейке присваиваем номер пятна в соседней ячейке
          c[b[i,j]-1]:= c[b[i,j]-1]+1//увеличиваем площадь пятна с нужным номером(у индекса массива С отнимаем 1, так как нумерация массива идёт с 0, а пятен с 1)
          end
        else if((j>0)and(b[i,j-1]<>0)) then //сверху
            begin
            b[i,j]:=b[i,j-1];
            c[b[i,j]-1]:= c[b[i,j]-1]+1
            end
          else if((i>0)and(j>0)and(b[i-1,j-1]<>0)) then //по диагонали
              begin
              b[i,j]:=b[i-1,j-1];
              c[b[i,j]-1]:= c[b[i,j]-1]+1
              end
            else // если соседние ячейки чисты, то
              begin
              k:=k+1;//увеличиваем кол-во пятен
              b[i,j]:=k;//присваеваем ячейке номер нового пянта
              c[b[i,j]-1]:=1;//площадь нового пятна 1
              end;
    end;
writeln(f2, k); //выводим количество пятен
qsort(0,k-1); // сортируем массив с (по возрастанию площади)
i:=0;
while(i<=k-1) do //проходим по всему массиву с
  begin
  l:=1;// хранит кол-во пятен с одинаковой площадью
  while ((c[i]=c[i+1])and(i<k-1)) do //если площадь этого пятна равна площади следующего то
    begin
    l:=l+1;
    i:=i+1;
    end;
  writeln(f2, c[i], ' ', l);//выводим площадь пятна и кол-во таких пятен
  i:=i+1;
  end;
close(f1);
close(f2);
end.