

Проект4. Количество путей в лабиринте

Карта лабиринта представляет собой квадратное поле размером $N \times N$. Некоторые квадраты этого поля запрещены для прохождения. Шаг в лабиринте – перемещение из одной разрешенной клетки к другой разрешенной клетке, смежной с первой по стороне. Путь – это некоторая последовательность таких шагов. При этом каждую клетку, включая начальную и конечную, можно посещать несколько раз.

Требуется написать программу, которая подсчитает количество различных путей из клетки (1, 1) в клетку (N, N) ровно за K шагов (то есть оказаться в клетке (N, N) после K-го шага).

Входные данные

Входной файл INPUT.TXT содержит в первой строке числа N и K, разделенные пробелом ($1 < N \leq 15$, $0 < K \leq 30$). Следующие N строк, по N символов в каждой, содержат карту лабиринта, начиная с клетки (1, 1). Символ «0» означает не запрещенную для прохождения клетку, а символ «1» - запрещенную. Начальная и конечная клетки всегда разрешены для прохождения.

Выходные данные

Выходной файл OUTPUT.TXT должен содержать количество возможных различных путей длины K. Во всех тестах это значение не будет превышать 2147483647.

№	INPUT.TXT	OUTPUT.TXT
1	3 6 000 101 100	5
2	2 8 01 10	0
3	3 6 0 0 0 1 1 1 0 0 0	0

Решение: Пусть $A[k, i, j]$ - количество способов попасть в клетку (i, j) ровно за k ходов.

За 0 ходов можно попасть только в клетку [1, 1], т.е. $A[0, 1, 1] = 1$. Для всех остальных $A[0, i, j] = 0$

	(i-1, j)	
(i, j-1)	(i, j)	(i, j+1)
	(i+1, j)	

Для $k \neq 0$ в клетку (i, j) (если она не запрещена) можно попасть из соседних клеток (i-1, j), (i+1, j), (i, j-1) и (i, j+1). Общее количество способов попасть в клетку (i, j) после k хода равно сумме количества ходов из соседних клеток на предыдущем ходе или 0, если клетка запрещена, то есть:

if $c[i, j] = 0$ then $a[k, i, j] := a[k-1, i, j-1] + a[k-1, i, j+1] + a[k-1, i-1, j] + a[k-1, i+1, j]$ else
 $a[k, i, j] := 0$;

В массиве A будем хранить только два последних хода.
 program labirint;

```

uses crt;
var a:array[0..1,0..20,0..20] of integer;
c:array[1..20,1..20] of 0..1;
n,k,i,j,k0,k1,m:longint;
begin
assign(input,'input.txt'); reset(input);
readln(n,k);
for i:=1 to n do
for j:=1 to n do if j<>n then read(c[i,j]) else readln(c[i,j]);
close(input);
{ -----начальные значения A----- }
for i:=1 to n do for j:=1 to n do
begin a[0,i,j]:=0;   a[0,0,j ]:=0;   a[0,i,0]:=0;end;
a[0,1,1]:=1;
{ -----sapolnenieA----- }
k0:=0;k1:=1;
for m:=1 to k do
begin
for i:=1 to n do for j:=1 to n do
if c[i,j]=0 then a[k1,i,j]:= a[k0,i,j-1]+a[k0,i,j+1]+a[k0,i-1,j]+a[k0,i+1,j]
else a[k1,i,j]:=0;
k0:=1-k0;k1:=1 - k1;
end;
{ -----otwet----- }
assign(output,'output.txt');rewrite(output);
if k1=1 then writeln( a[0,n,n]) else writeln( a[1,n,n]);
close(output);

```

end