

Робот

В исследовательской лаборатории фирмы Robots&Co разработали новую модель робота. Главной особенностью данной модели робота является то, что он работает по заранее заданной программе, в которой могут присутствовать команды: сделать шаг на Юг, на Север, на Восток или на Запад. Робот исполняет программу строго последовательно и, дойдя до конца программы, останавливается. Специалисты из Robots&Co заинтересовались вопросом, сколько существует различных программ, состоящих из K инструкций, таких, что робот, выйдя из начала координат, придет в точку с координатами (X, Y) . Оси координат располагаются параллельно сторонам света, и единица измерения, соответствует одному шагу робота. Напишите программу, которая дает ответ на этот вопрос.

Формат входных данных

Во входном файле находятся три числа K, X и Y ($0 \leq K \leq 16, |X|, |Y| \leq 16$), разделенные пробелами.

Формат выходных данных

В выходной файл ваша программа должна поместить одно число — количество программ для робота.

Робот

В этой задаче мы впервые сталкиваемся с функцией трех переменных:

$F(X, Y, Z)$ - показывает количество маршрутов длины Z , приводящих в клетку (X, Y) .

К сожалению, такая структура данных как Array $[-16..16, -16..16, 0..16]$ of LongInt; в память не помещается. Забудем пока об этой трудности и напишем рекурсивную часть. Для ответа на поставленный в задаче вопрос можно посчитать число маршрутов длины $Z - 1$ в четыре клетки, смежных с заданной, а результаты сложить, не забывая случаи клеток на границе.

Граничными условиями будут $F(0, 0, 0) = 1$. Действительно, если робота не двигать, то он может оказаться только в начале координат.

Достаточно хранить только 2 квадратные матрицы размером 33, - одна несет данные в предыдущий момент времени, а вторая вычисляется для текущего с использованием первой матрицы. После окончания расчета данные первой уже не нужны, ей присваиваем значение второй матрицы, увеличиваем счетчик времени и т.д. Ясно, что такую идею можно реализовать только итеративно.

```
for z := 1 to k do
begin
  Way2 := Way1;
  for i := -16 to 16 do
    for j := -16 to 16 do
      begin
        s := 0;
        if i <> -16 then s := s + Way2[i - 1, j];
        if i <> 16 then s := s + Way2[i + 1, j];
        if j <> -16 then s := s + Way2[i, j - 1];
        if j <> 16 then s := s + Way2[i, j + 1];
        Way1[i, j] := s;
      end;
    end;
  end;
```

Текст программы

```
var way1, way2: array[-16..16, -16..16] of longint;
i, j, s, k, z, x, y: longint;
begin readln(k); way1[0, 0] := 1;
for z := 1 to k do
begin
  Way2 := Way1;
  for i := -k to k do
  for j := -k to k do
  begin
```

```

    s := 0;
    if i <> -k then s := s + Way2[i - 1, j];
    if i <> k then s := s + Way2[i + 1, j];
    if j <> -k then s := s + Way2[i, j - 1];
    if j <> k then s := s + Way2[i, j + 1];
    Way1[i, j] := s
end ;
for i := -k to k do for j := -k to k do if j<>k then write(way2[i,j]:3) else
writeln(way2[i,j]:3);
writeln('-----');
for i := -k to k do for j := -k to k do if j<>k then write(way1[i,j]:3) else
writeln(way1[i,j]:3);
readln;
end; repeat readln(x,y);until (abs(x)<=k) and (abs(y)<=k);if (x=0) and(y=0) then writeln(k)
else writeln(way1[x,y]); readln;
end.

```