

## Незнайка и ререновация. 7-10 классы

Знайка побывал на экскурсии в Солнечном городе и ему там очень понравилось. Вернувшись домой, он решил провести реновацию Цветочного города – расширить его и полностью перестроить по примеру Солнечного города. По плану Знайки обновлённый Цветочный город стал квадратом из  $N * N$  одинаковых квадратных участков. На каждом участке был выстроен один домик коротышки. Домики были перенумерованы согласно бустрофедоническому способу, названному так из-за схожести формой борозды, оставляемой при вспашке поля плугом, который тянет бык.

Коротышки согласились с реновацией, предложенной Знайкой. Один лишь Незнайка решил разузнать, как на самом деле обстоят дела в Солнечном городе. Оказалось, что домики там нумеровались спиральным способом, а вовсе даже не бустрофедоническим. Домик с номером 0 (ноль) располагался в северо-западном углу (на плане это верхний левый угол). От него нумерация шла вдоль горизонтальной стороны квадрата по направлению вправо (или по ходу часовой стрелки). Достигнув правого края, нумерация продолжалась вдоль вертикальной стороны квадрата по направлению вниз. Достигнув нижнего края, она следовала вдоль горизонтальной стороны квадрата по направлению влево. Дальнейший её ход замыкал обход периметра квадрата по часовой стрелке. Потом следовал ещё один виток спирали по периметру внутреннего квадрата  $(N - 2) * (N - 2)$ . И так далее. Например, при  $N = 6$  результат нумерации был бы таким:

0	1	2	3	4	5
19	20	21	22	23	6
18	31	32	33	24	7
17	30	35	34	25	8
16	29	28	27	26	9
15	14	13	12	11	10

Эти сведения Незнайка тут же сообщил Знайке и стал требовать соблюдения правил нумерации домиков Солнечного города. Знайка обещал рассмотреть новую реновацию – ререновацию, как он её назвал, – только в том случае, если будет надёжный способ узнать номер каждого домика по его расположению. В который раз Незнайке нужна помощь с составлением программы.

Требуется написать программу, которая считывает  $N$  – количество участков вдоль одной стороны города. Затем программа считывает  $I$  – номер вертикального ряда домиков и  $J$  – номер горизонтального ряда домиков. Рядам присвоены номера, начиная с 1, слева направо для вертикальных рядов и сверху вниз – для горизонтальных. На пересечении рядов, с номерами  $I$  и  $J$  находится домик, номер которого в спиральной нумерации следует найти. Программа находит и выводит результат – искомый номер домика. Например, если  $N = 6$ , как выше, и  $I = 4$ , а  $J = 2$ , то искомым номером будет 22.

0	1	2	3	4	5
19	20	21	22	23	6
18	31	32	33	24	7
17	30	35	34	25	8
16	29	28	27	26	9
15	14	13	12	11	10

**Формат ввода:** В первой строке вводится натуральное число  $N$  в десятичной записи:  $0 < N < 1001$ . Во второй строке вводится натуральное число  $I$ :  $0 < I < N + 1$ . В третьей строке вводится натуральное число  $J$ :  $0 < J < N + 1$ .

**Формат вывода:** Выводится десятичная запись искомого номера домика согласно спиральной нумерации. При выводе незначимые нулевые разряды не выводятся.

Ввод примера №1:		Ввод примера №2:		Ввод примера №3:
1		2		6
1		1		4
1		2		2
Вывод примера №1:		Вывод примера №2:		Вывод примера №3:
0		3		22

## Решение

Сначала определяется  $K$  – номер витка спирали, во время прохода по которому будет пронумерован домик на пересечении рядов с номерами  $I$  и  $J$ . Он равен увеличенному на 1 расстоянию от домика до ближайшего края квадрата (вертикального или горизонтального – того, что окажется ближе). В примере №3  $K = \min(I, J, (N + 1 - I), (N + 1 - J)) = 2$ . Номер  $M$ , назначенный домику в начале витка, определяется как сумма первых  $K - 1$  элементов последовательности:  $4 * (N - 1), 4 * (N - 3), \dots, 12$  для чётных  $N$  или  $4 * (N - 1), 4 * (N - 3), \dots, 8$  для нечётных  $N$ . Выведем формулу для  $M$ . При  $K = 1$   $M = 0$ . При больших  $K$   $M = 4 * (N - 1) - 4 * 2 * 0 + 4 * (N - 1) - 4 * 2 * 1 + \dots + 4 * (N - 1) - 4 * 2 * (K - 2) = 4 * ((N - 1) * (K - 1) - (K - 2) * (K - 1)) = 4 * (K - 1) * (N - K + 1)$ . В примере №3  $M = 4 * 1 * 5 = 20$ . Имеются 4 случая расположения домика:

- 1) Он находится на начальном участке витка – в горизонтальном отрезке, проходимом слева направо (в примере №3 именно этот случай,  $K = J = 2$ ). Искомый номер находится как  $M + (I - K) = 20 + 2 = 22$ .
- 2) Он находится на втором участке витка – в вертикальном отрезке, проходимом сверху вниз (в таком случае  $N + 1 - I = K$ ). Искомый номер находится как  $M + (N + 1 - 2 * K) + (J - K) = M + N + J - 3 * K + 1$ .
- 3) Он находится на третьем участке витка – в горизонтальном отрезке, проходимом справа налево (в таком случае  $N + 1 - J = K$ ). Искомый номер находится как  $M + 3 * (N + 1 - 2 * K) - (I - K) = M + 3 * N - I - 5 * K + 3$ .
- 4) Он находится на последнем участке витка – в вертикальном отрезке, проходимом снизу вверх (в таком случае  $K = I$ ). Искомый номер находится как  $M + 4 * (N + 1 - 2 * K) - (J - K) = M + 4 * N - J - 7 * K + 4$ .

## Код возможного решения

```
program SPIRAL710(input, output);
var N, I, J, K: word;
    M, RESULT: longword;
function MIN4(A1, A2, A3, A4: word): word;
begin
    if (A1 > A2) then A1 := A2;
    if (A3 > A4) then A3 := A4;
    if (A1 > A3) then MIN4 := A3 else MIN4 := A1
end;

begin
    readln(N);
    readln(I);
    read(J);
    K := MIN4(I, J, (N+1-I), (N+1-J));
    M := 4 * (K - 1) * (N - K + 1);
    if (K = J) then RESULT := M + I - K
    else if (N + 1 - I) = K then RESULT := M + N + J - 3 * K + 1
    else if (N + 1 - J) = K then RESULT := M + 3 * N - I - 5 * K + 3
    else RESULT := M + 4 * N - J - 7 * K + 4;
    write(RESULT)
end.
```