R2

By definition, the arithmetic mean S=(r1+r2)/2. Therefore, r2=2*s-r1.

ABC

We first sort the three input numbers to find which of them are actually A, B and C. We then input the desired order and output the three numbers in that order.

KOLONE

One approach is to simulate the ants' behaviour second by second.

A more efficient approach directly calculates the final position of each ant.

Let's label the ants with numbers within their rows, starting with 0.

We know the initial (T=0) position P of each ant: P=N1-i-i1 for the first (left) row and P=N1+I for the other (right).

There are three cases:

1. If T is less than i, then the ant will not jump over any other ants and will remain in his initial position.
2. If T is greater than or equal to i+(the number of ants in the other row), then the ant will jump over all of the ants in the other row.
3. Otherwise, the ant's position will change by (T-i).

SJECISTA

Imagine that the polygon's vertices are labeled 0 to N-1 consecutively. Consider a diagonal from vertex 0 to some vertex i. Vertices 1 through i-1 are on one side of the diagonal, vertices i+1 through N-1 on the other. The diagonal intersects only those diagonals which have one vertex on one side and the other on the other side. Thus there are (i-1) * (N-1-i) diagonals intersecting it.

We count all diagonals with one vertex being vertex 0 and multiply the number of diagonals by n (we could have labeled any vertex 0 and would get the same result). We counted each intersection on one diagonal twice (once from each vertex on a single diagonal) and each intersection a total of four times (because it lies on two diagonals). Hence we divide the result by 4.

STOL

Suppose one dimension of the table is fixed. For example, let the table start in column s1 and end in column s2. For the table to span a certain row, all squares between s1 and s2 (inclusive) in that row must be free.

For each row we can determine if there are any blocked squares between columns s1 and s2 in constant time: we preprocess the apartment and calculate the number of blocked squares left of each square. Call this number sum(row, column). After calculating this, the expression sum(row, s2) – sum(row, s1-1) tells us how many squares between s1 and s2 are blocked (if this number is zero, then the table can be placed in that row).

A single pass finds the largest consecutive number of rows that don't have walls between s1 and s2, giving the largest solution for the assumed columns s1 and s2. Exploring all possibilities for s1 and s2 we choose the one that gives the maximum perimeter.

The number of possibilities is proportional to N^2. Using the above algorithm, the total running time is proportional to N^3.

STRAZA

The trenches are represented by line segments. Suppose no two line segments overlap. Then any three segments, such that each segment intersects both of the other two and the three intersections are distinct, form a solution – we can place the guards on the three intersections.

Because some line segments may overlap we need to preprocess the input, merging line segments that overlap until there are none left.

The time complexity of the solution is $O(N^3)$.