**Problem «Ceremony**»

The main topics of task 1 " Ceremony" are: linear search options, calculating the length of the segment at its ends. A possible partial solution is based on a complete iteration of the options.

Let the length of the smaller side be *x* and the length of the larger side be *y*. then note that the following constraints must be satisfied:

*y* ≥ *x*

*x*∙y ≥ *A*, therefore *y* ≥ *A* / *x*

2∙(*x* + *y*) ≥ *C*, therefore *y* ≥ *C* / 2 – *x*

Thus, the minimum suitable value of *y* is:

*ymin* = max{*x*, *A* / *x*, *C* / 2 – *x*},

and the last two values must be rounded up.

*x∙y* ≤ *B*, therefore *y* ≤ *B* / *x*

2∙(*x* + *y*) ≤ *D*, therefore *y* ≤ *D* / 2 – *x*

Thus, the maximum suitable value of y is:

*ymax* = min{*B* / *x*, *D* / 2 – *x*},

and this time the values must be rounded down.

Next, we iterate over the smaller side of the hall and note that the search can be carried out to the square root of *B*. for each value of x, calculate the minimum and maximum values of y and add to the answer max{0, *ymax* – *ymin* + 1}.

When solving this subtask, you can touch both sides of the hall.

Below is the corresponding program-the solution of the problem in Pascal.

    read(a, b, c, d);

    ans := 0;

    for x := 1 to d div 2 do begin

        if x \* x > b then

            break;

        miny := x;

        if (c + 1) div 2 - x > miny then

            miny := (c + 1) div 2 - x;

        if (a + x - 1) div x > miny then

            miny := (a + x - 1) div x;

        maxy := d div 2 - x;

        if b div x < maxy then

            maxy := b div x;

        if maxy >= miny then

            ans := ans + (maxy - miny + 1);

    end;

    writeln(ans);

A similar approach, but with simpler formulas, is obtained if you apply the following technique. We abandon the lower constraints and solve the problem for the number of rooms with an upper bound for the area *B* and an upper bound for the perimeter *D*. we denote the number of such rooms as f*(B, D)*. Now, to get the correct number of rooms, you can apply the inclusion-exclusion formula:

*ans* = f(*B*, *D*) – f(*B*, *C* – 1) – f(*A* – 1, *D*) + f(*A* – 1, *C* – 1)