

C. Chocolate Bar

Time limit: 2s

Memory limit: 256 MB

You have a rectangular chocolate bar consisting of $n \times m$ single squares. You want to eat **exactly** k squares, so you may need to break the chocolate bar.

In one move you can break any single rectangular piece of chocolate in two rectangular pieces. You can break only by lines between squares: horizontally or vertically. The cost of breaking is equal to square of the break length.

For example, if you have a chocolate bar consisting of 2×3 unit squares then you can break it horizontally and get two 1×3 pieces (the cost of such breaking is $3^2 = 9$), or you can break it vertically in two ways and get two pieces: 2×1 and 2×2 (the cost of such breaking is $2^2 = 4$).

For several given values n , m and k find the minimum total cost of breaking. You can eat exactly k squares of chocolate if after all operations of breaking there is a set of rectangular pieces of chocolate with the total size equal to k squares. The remaining $n \cdot m - k$ squares are not necessarily form a single rectangular piece.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 40910$) — the number of values n , m and k to process.

Each of the next t lines contains three integers n , m and k ($1 \leq n, m \leq 30$, $1 \leq k \leq \min(n \cdot m, 50)$) — the dimensions of the chocolate bar and the number of squares you want to eat respectively.

Output

For each n , m and k print the minimum total cost needed to break the chocolate bar, in order to make it possible to eat exactly k squares.

Examples

input
4
2 2 1
2 2 3
2 2 2
2 2 4
output
5
5
4
0

Note

In the first query of the sample one needs to perform two breaks:

- to split 2×2 bar into two pieces of 2×1 (cost is $2^2 = 4$),
- to split the resulting 2×1 into two 1×1 pieces (cost is $1^2 = 1$).

In the second query of the sample one wants to eat 3 unit squares. One can use exactly the same strategy as in the first query of the sample.

