Minimum Sum

Suppose you have a list of integers, and a ***move*** is defined as taking one of the integers from the list and replacing it with its square root, rounded down to the nearest integer.

Given an integer l and an integer k, start with the array [1, 2, 3, ..., l] and find the minimal sum of the array after k *moves*.

Example

For l = 5 and k = 2, the output should be
squareRoots(l, k) = 10.

We start with [1, 2, 3, 4, 5].
After square rooting 5 to get [1, 2, 3, 4, 2] and then square rooting 3 to get[1, 2, 1, 4, 2], we end up with a sum of 10.

Constraints:

$1 \leq l \leq 10^4$

$1 \leq k \leq 10^4$

T=10000

Input :

The first line contains T the number of test cases followed by 2*T lines containing l and k.

Output:

For every test case, output one line containing an integer, i.e. the minimal possible sum.

Sample Input:

```
2

5

2

2327

4895
```

Sample Output:

```
10

10647
```