Using the definitions

```cpp
typedef vector<int> VI;
typedef vector<VI> VVI;
```

implement a function

```cpp
int k_esim(int k, const VVI& V);
```

to return the $k$-th global element (starting at one) of the elements in the vector of vectors $V$.

Let $n = V.size()$. For every $0 \leq i < n$, $V[i]$ is sorted increasingly. Furthermore, there are no repeated elements in $V$.

For example, if $k = 5$, $n = 3$, and the three vectors are

| $V[0]$ | 1 | 2 | 10 | 15 |
| $V[1]$ | -5 | -3 | 12 |
| $V[2]$ | 0 | 3 | 4 | 6 | 20 |

then the answer is 2, which is the fifth smallest element inside all the vectors.

Let $m = \sum_{i=0}^{n-1} V[i].size()$. Assume that $k$ is between 1 and $m$, that $n$ is between 2 and 500, and that some $V[i]$ can be empty. If needed, you can implement auxiliary procedures. Take into account that, for the “large” test cases, $k = \Theta(n)$ and $m = \Theta(n^2)$. The expected solution in this case has cost $\Theta(n \log n)$.

**Observation**

You only need to submit the required procedure; your main program will be ignored.