Implicit SLAE

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

The task is simple: you have to solve a system of linear algebraic equations $Ax = b$ with residual no more than $\varepsilon = 10^{-9}$. And the matrix A is very nice: **non-singular, symmetric and strictly diagonally dominant**. A piece of cake, you might think. The catch: you do not have A in any explicit form. You can only get the result of its multiplication with the vector.

As in the previous problem, you have a number of blackbox functions through which you work with the SLAE:

- `void blackbox_init()` – initializes the internal blackbox data structures. Should be called in the very beginning of the program! No other blackbox function should be called before it, and no reading from the `stdin` shall be made (or at least, as they say, "be kind, `rewind`").
- `int blackbox_size()` – returns the number of equations (which is equal to the number of unknowns) of the system. The number of equations lies between 10 and 10000 (inclusive).
- `void blackbox_mult(const double *x, double *out)` – compute the product of A and vector `x`, write the results to `out`. The pointers `x` and `out` should point to *different* chunks of memory of size at least `blackbox_size() * sizeof(double)` bytes each.
- `void blackbox_rhs(double *b)` – write the right-hand side of the SLAE (*i.e.,* vector b) to the array `b`. The pointer `b` should point to the chunk if memory of size at least `blackbox_size() * sizeof(double)` bytes.
- `void blackbox_submit(double *solution)` – write the result of the program. The array `solution` should contain the solution to the SLAE: `blackbox_size()` values of type `double`. This should the last function to be called by your program (besides `return 0;`).

Input

Use `void blackbox_init()`. You should not perform any I/O yourself.

Output

Use `void blackbox_submit(double *solution)`. You should not perform any I/O yourself.