Notice

If you are unable to submit a solution via CMS, please do as follows:

- Put your submission files in a directory named `submit_<task name>` on the desktop (the directory should already exist) **before the contest ends**.
  - For output-only problems, your files must be named `01.out`, `02.out`, ..., `10.out`.
  - For other problems, there must be **exactly one** file in that folder, containing your source code.
- Ask your team leader to submit an appeal.

If you think you should be given extra time, please do as follows:

- Send a clarification request (through the contest system or on paper) as soon as possible.
- Do not leave your desk or talk to other contestants after the contest ends.

Each task has an attachment package that you can download from the contest system.

There are some "output-only" tasks, for which:

- The attachment package contains input test cases and example test cases. Each test case is a separate subtask.
- You may submit multiple output files as a zip file. For this purpose, your output files should be named `??.out`, where `??` is the test case number (e.g., `03.out`). You can zip multiple files using the following command: `zip output.zip *.out`
- You may make up to 100 submissions for output-only tasks. In each submission, you may submit the output files for any subset of the test cases.

For other tasks:

- The attachment package contains sample graders, sample implementations, example test cases, and compile scripts.
- You may make up to 50 submissions for each task, and you have to submit exactly one file in each submission.
- The name of the file that you should submit is given in the task statement header. It should implement the procedures described in the task statement using the signatures provided in the sample implementations.
- You are free to implement other procedures.
- Your submissions must not read from the standard input, write to the standard

output, or interact with any other file. However, they may output to the standard error stream.
- Your submissions **must not call `exit()` or `System.exit()`**.
- When testing your programs with the sample grader, your input should match the format and constraints from the task statement, otherwise, unspecified behaviors may occur.
- In sample grader inputs, every two consecutive tokens on a line are separated by a single space, unless another format is explicitly specified.
- When you test your code on your local machine, we recommend you to use scripts in the attachment package. Otherwise, especially in C++, make sure to add `-std=gnu++14` option to compile.

Conventions

The task statements specify signatures using generic type names `void`, `int`, `int64`, and `int[]` (array).

In each of the supported programming languages, the graders use appropriate data types or implementations, as listed below:

Language	void	int	int64	int[]	length of array a
C++	`void`	`int`	`long long`	`std::vector<int>`	`a.size()`
Java	`void`	`int`	`long`	`int[]`	`a.length`

Limits

Task	Time limit	Memory limit
line	output-only	output-only
vision	1 sec	1024 MB
walk	4 sec	1024 MB