



## Implementation Notice

- . You have to submit exactly one file (file name given in the problem statement) ◦
  - This file implements the subprograms described in the task statement using ◦
  - signatures provided in the sample implementation ◦
  - . These subprograms must behave as described in the task statement ◦
- . You are free to implement other subprograms (functions, procedures, methods) ◦
  - Your submissions must not interact in any way with standard input/output ◦
  - stream, nor with any other file. In particular, if your program outputs anything ◦
  - to standard output stream, its grading outcome on this test will be SV (Security ◦
  - . Violation). You may output anything to standard error stream

## Conventions

The task statements use the word *array* and the Implementation details sections use the type `int[]`. Depending on the programming language, the graders use the following types in place of `int[]` (and for simplicity, we call all of them arrays)

- , `++std::vector<int>` in C ◦
- , `int*` in C ◦
- , `array of longint` in Pascal ◦
- . `int[]` in Java ◦

The Implementation details sections use the type `int64`. It corresponds to 64-bit signed integer type

- , `long long` in C++/C ◦
- , `int64` in Pascal ◦
- . `long` in Java ◦

## Limits

Problem	Time Limit	Memory Limit
Detecting Molecules	1 second	2 GB
Roller Coaster Railroad	2 seconds	2 GB
Shortcut	2 seconds	2 GB