



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 and the Implementation details sections use some generic type names, in particular:

- the name *array* and the corresponding type `int[]`
- the type `int64`
- the type `string`
- the type `boolean`

In each of the supported programming languages the graders use the appropriate data types from that language, as listed below:

| Language | array | int64 | string | boolean |
|----------|-------------------------------------|------------------------|--------------------------|----------------------|
| C++ | <code>std::vector<int></code> | <code>long long</code> | <code>std::string</code> | <code>bool</code> |
| C | <code>int*</code> | <code>long long</code> | <code>char*</code> | <code>int</code> |
| Pascal | <code>array of longint</code> | <code>int64</code> | <code>string</code> | <code>boolean</code> |
| Java | <code>int[]</code> | <code>long</code> | <code>String</code> | <code>boolean</code> |

Limits

| Problem | Time Limit | Memory Limit |
|--------------------------|------------|--------------|
| Paint By Numbers | 2 seconds | 2 GB |
| Unscrambling a Messy Bug | 2 seconds | 2 GB |
| Aliens | 2 seconds | 2 GB |