

# DAY 0 - OVERVIEW

Name of the task:	<b>UNIQUE NUMBER</b>
Input file name:	<code>unique.in</code>
Output file name:	<code>unique.out</code>
Time limit for one test case (depends on test case):	1 or 2 seconds
Source code file size limit:	1 MB
Memory limit:	64 MB
Maximum total points:	100
Comment which must be in source code first four lines if program is in <b>PASCAL</b> :	<pre>{ task: unique lang: pascal }</pre>
Comment which must be in source code first four lines if program is in <b>C</b> :	<pre>/* task: unique lang: c */</pre>
Comment which must be in source code first four lines if program is in <b>C++</b> :	<pre>/* task: unique lang: c++ */</pre>
Precondition for program to be tested:	Source code compiles without errors and passes example test case given in task description

# UNIQUE NUMBER

DAY 0 (21-APR-2004)  
ENGLISH VERSION



There is given sequence of positive integers. It is known that one of numbers in this sequence is present only once, but others exactly  $k(k>1)$  times each.

Write a program which finds this number which is present in sequence only once!

## *Input*

In the first line of text file `unique.in` one positive integer  $n(n \leq 2000001)$  is given - total number of sequence elements.

Each of the next  $n$  file lines contains one sequence element.  $i$ -th sequence element is given in the file's  $i+1$ -st line. Any element does not exceed 2147483647.

## *Output*

In the only line of text file `unique.out` program must output one integer - number which is present in sequence only once.

## *Example*

<code>unique.in</code>	<code>unique.out</code>
13	295
537	
295	
210	
413	
413	
210	
413	
210	
413	
210	
537	
537	
537	