

Problem 5. Fix the heap 32-bit

Input file: standard input stream
Output file: standard output stream
Time limit: 3 seconds
Memory limit: 256 megabytes

Petya wrote a heap for dynamic memory allocation in his own programming language Tse. Here is how it works.

A correct heap is a continuous segment of memory consisting of N cells. Each cell contains an unsigned **32-bit** integer, which can have any value **from 0 to 4 294 967 295** inclusive. The heap is split into B sub-segments called blocks. Each of the memory cells belongs to strictly one block.

Each block consists of two or more cells. The first and the last cells of a block contain the effective size of the block, which is the number of cells in it, excluding the first and last cells. The rest of the block's cells can contain arbitrary data regardless of whether the block is free or occupied.

Tse is a very low-level language, and its users often mess everything up and corrupt the heap by inadvertently writing their data into wrong cell. Users are asking Petya to come up with a feature of recovering heap integrity after such incidents. The recovery code must analyze the contents of N cells of the memory segment where the heap is supposed to be located, and change the contents of several memory cells in such a manner that the segment becomes a correct heap. The number of modified cells must be minimal.

Interaction Protocol

The number of memory cells can be very large, and your program won't be able to read the whole segment. For this reason, this is an interactive task. Instead of file input-output, you will be working with a special program called interactor. Interaction with this program is performed by means of the standard input-output streams.

At the start, your program receives an integer N in the standard input stream, being the number of memory cells in the analyzed segment ($2 \leq N \leq 2^{32}$). Next, your program can make queries to discover the contents of various memory cells. To make a query, print the word `read` into the standard output stream, followed by a space-separated integer A , being the address/index of the cell ($0 \leq A < N$). In response, a single integer V will be sent to the standard input stream, being the value stored in the A -th memory cell ($0 \leq V < 2^{32}$).

Eventually, your program must print the answer to the problem instead of yet another query. In the first line of the answer, print the word `fix`, followed by a space-separated integer K , being the number of cells that must be changed ($K \geq 0$). In the remaining K lines, print the suggested changes, one per line. For each change, print two integers: A being address/index of the cell that must be changed and V being new value written into this cell ($0 \leq A < N$, $0 \leq V < 2^{32}$).

It is guaranteed that there exists an optimal answer resulting in a heap with no more than 25 000 blocks. Your program is allowed to make no more than 120 000 queries, otherwise the solution will get **Wrong Answer**.

Make sure you print the end-of-line symbol and clear the output stream buffer (the `flush` command of the language) after every printed query. Otherwise the solution can get **Timeout**. All numbers are written in decimal form in text format.

Examples

For reading convenience, commands in the example are separated by lines with the minus symbol. Your program must **not** print any minuses.

standard input stream	standard output stream
11	-
-	read 0
2	-
-	read 1
170	-
-	read 2
187	-
-	read 3
204	-
-	read 4
0	-
-	read 5
221	-
-	read 6
3	-
-	read 7
171	-
-	read 8
172	-
-	read 9
173	-
-	read 10
3	-
-	fix 2
-	3 2
-	5 0

Example explanation

The example fully repeats the second example from another problem «Fix the heap 8-bit»: both the contents of the memory cells and the suggested recovery fixes are the same. Illustration:

