

Notes.

- This is a bonus assignment.
- If you have already sent us your AVL code and you manage to get full marks on Automarker, you will get 2 bonus grade points. If you manage to get full marks on Automarker and you did not send us your code earlier, you will get 1 bonus grade point. **Note:** The point of the exercise is to implement your own AVL tree and you will only get bonus points if you do that!
- You can use Java or C++. Libraries such as TreeSet, HashSet, etc. are **not** allowed.
 - **Java:** Please use FastScanner class for reading the input.
 - **C++:** If you are using std::cin and std::cout (which you should), make sure your main function starts with the following:

```
ios_base::sync_with_stdio(false);  
std::cin.tie(NULL);  
std::cout.tie(NULL);
```

- There is a limit of 5 submission attempts for this problem!
- Please do not share your code with other students!

1 AVL

Problem. You start with an empty set, and you are given a sequence of 1000000 operations. Each operation is of the form 0 x , in which case you are asked to put x into the set, or 1 x , in which case you should output 0 if x is currently not in your set, and 1 otherwise. You can assume that x is a 32-bit integer.

	Input:	Output:
	0 0	1
	0 1	0
	0 2	1
Example.	0 3	
	0 4	
	0 5	
	1 1	
	1 6	
	0 6	
	1 6	

Explanation of the example. Note that this example only has 10 operations. The real test data will contain exactly 1000000.

The first 6 operations are adding numbers 0, ..., 5 to the set. The next operation is asking whether 1 is in the set, and the answer is yes (thus we output 1). The next operation asks whether 6 is in the set, and the answer is no (output 0). We then add 6 to the set and again ask if 6 is the set. This time it is, so we output 1.