### SIENA COLLEGE

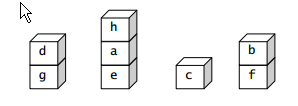
**25th Annual**

### High School Programming Contest

##### April 20, 2012

###### **Problem #6: Blocks**

Background Information: A classic problem in robotics is to program a robot arm that can pick up blocks lying on a table. In this problem we assume all blocks are of the same size. The relationship of their relative position is described by predicates. For example, consider the following block world (ignore the arrow)



It can be represented by the following predicates:

ontop(d, g) // block d is on top of block g

ontop(c, nil) // c is not on top of any block

ontop(a, e)

ontop(b, f)

ontop(h, a)

Note that if a block (such as g, e, and f) is not on top of any other block, it is right on the table. If a block is by itself, we use a predicate like ontop(c, nil) to indicate that it is not on top of any block. There will be one predicate per line in the input. There is no particular order in listing the predicates. Block names will all be a single lowercase letter; nil in lowercase will be used for singleton blocks.

Your program should output the height of the tallest stack (or stacks in case of ties) of blocks. In our example, the height of the tallest stack is 3.

###### Programming Problem:

Input: A positive integer *N* on its own line followed by N lines of predicates of

the form Letter1 Letter2, where Letter1 is on top of Letter2 or nil.

Output: The height of the largest stack.

###### Example 1: Input: **5**

**d g**

**c nil**

**a e**

**b f**

**h a**

###### Output: 3

###### Example 2: Input: **5**

**a b**

**c d**

**e f**

**g nil**

**b c**

###### Output: 4

###### Example 3: Input: **7**

**a nil**

**c nil**

**e nil**

**g nil**

**b d**

**f b**

**x y**

###### Output: 3

###### blocks.jpg