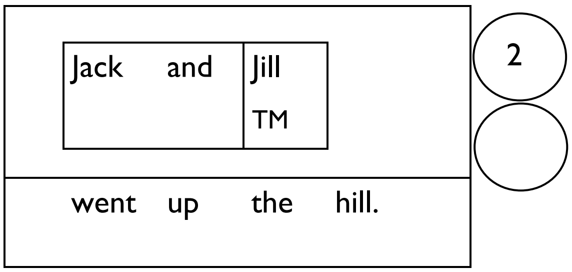
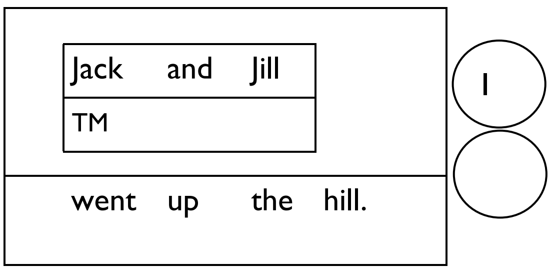
(H) Twodee (1/10)

H1. Our boxes below indicate just the phrases, as revealed by the Twodee layout, that are key to determining the choice of meaning. Overlapping boxes are used when the Twodee layout is "unclear" and allows multiple possible meanings.

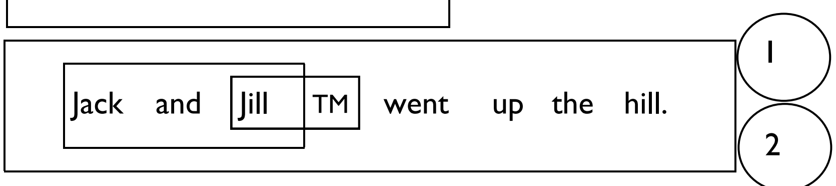
A.



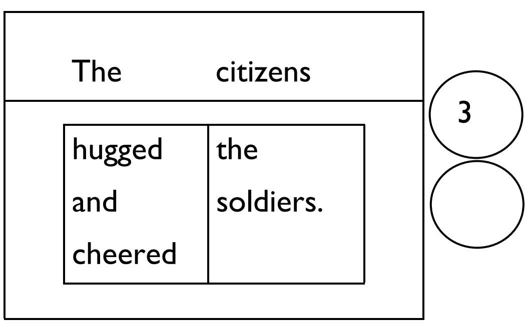
1. The trademarked group known as “Jack and Jill” ascended the hill.



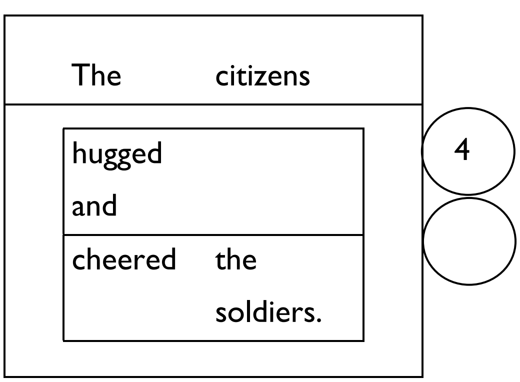
2. Jack ascended the hill with his trademarked companion, “Jill.”



B.



3. The citizens hugged the soldiers and cheered the soldiers.

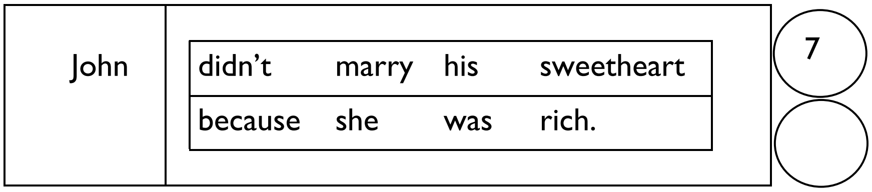


4. The citizens hugged each other and cheered the soldiers.

5. The citizens hugged the soldiers and cheered.

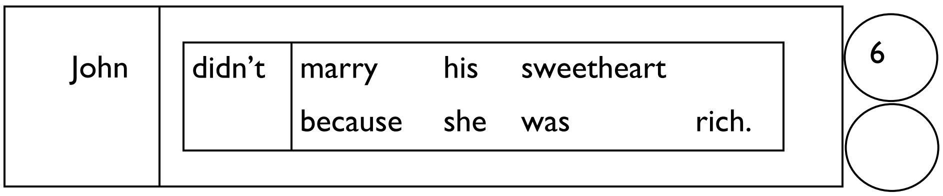
(H) Twodee (2/10)

C.

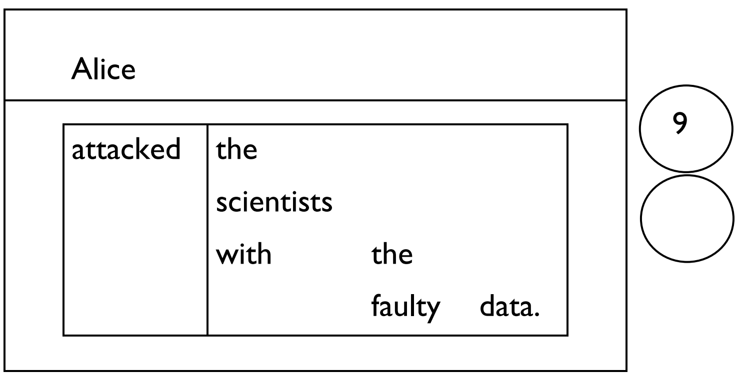


6. It's not because his sweetheart was rich that John married her.

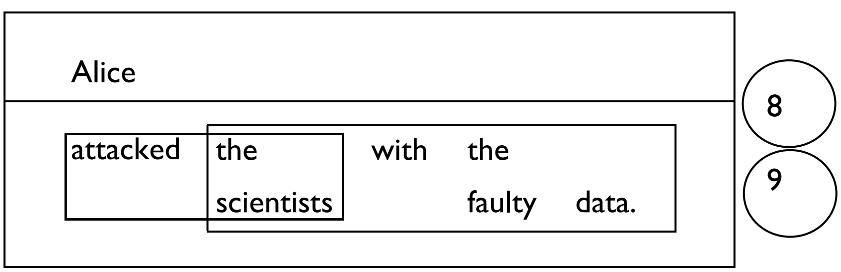
7. Because his sweetheart was rich, John didn't marry her.



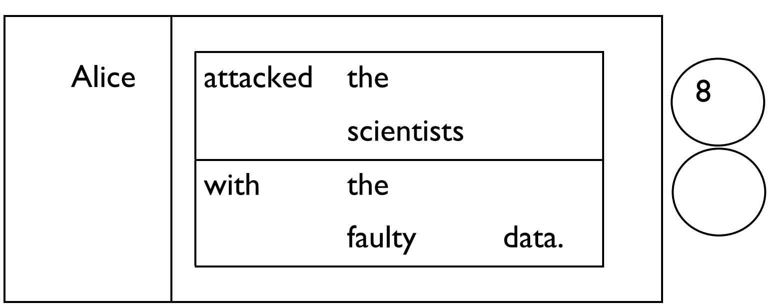
D.



8. Alice used the faulty data to attack the scientists.

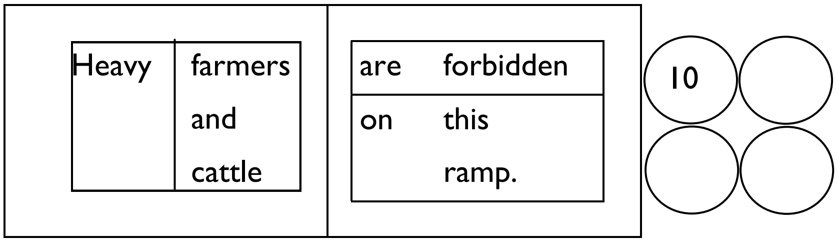


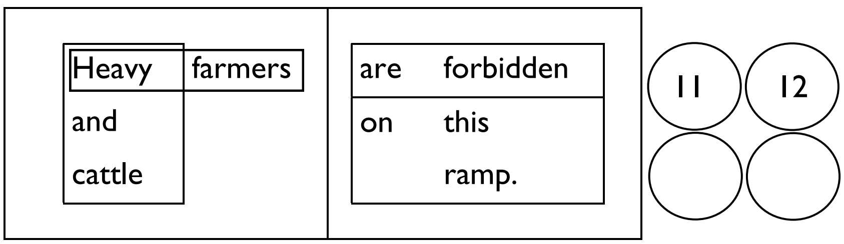
9. Alice attacked the scientists who had faulty data.



(H) Twodee (3/10)

E.





10. Heavy farmers and heavy cattle are forbidden on this ramp.

11. Heavy farmers and cattle farmers are forbidden on this ramp.

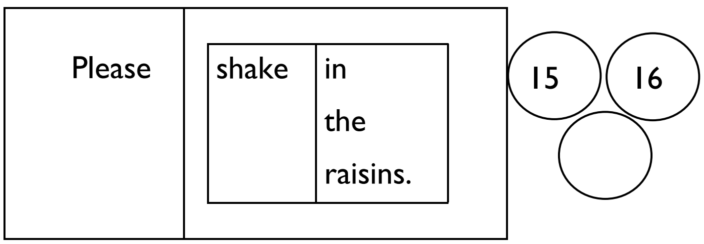
12. Cattle and heavy farmers are forbidden on this ramp.

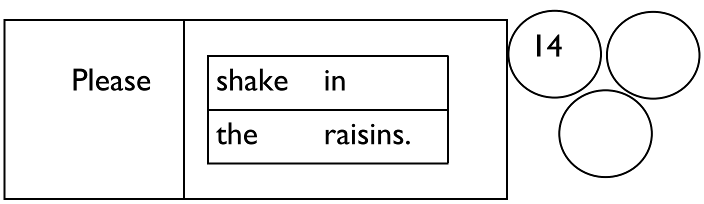
13. Cattle and heavy farmers are on this forbidden ramp.

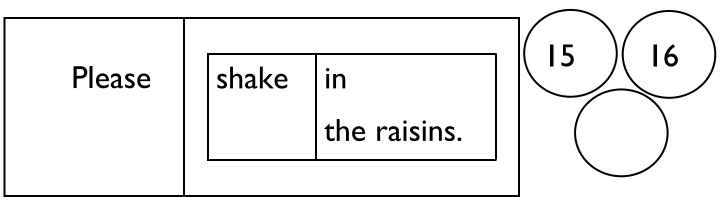
(H) Twodee (4/10)

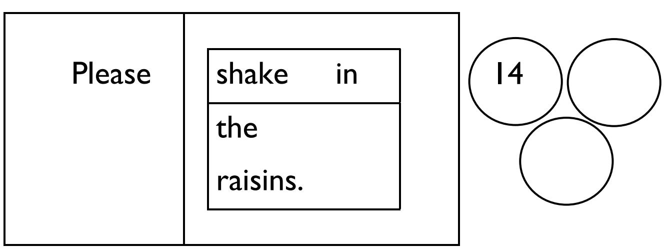
F. Twodee sentences #1 and #3 have the same meanings, as they are merely different written arrangements of the same phrases. Similarly for #2 and #4.

Twodee sentences #1 and #3 have meaning 16 if used in a context where you already wanted to shake yourself, but meaning 15 if used in a context where you didn't. Since the same sentence structure would be used for both contexts, the Twodee writing system cannot be used to distinguish between 15 and 16, any more than it can be used to distinguish whether the raisins are dark raisins or golden raisins. Twodee can clarify sentence structure but can't remove every last vestige of ambiguity from language.









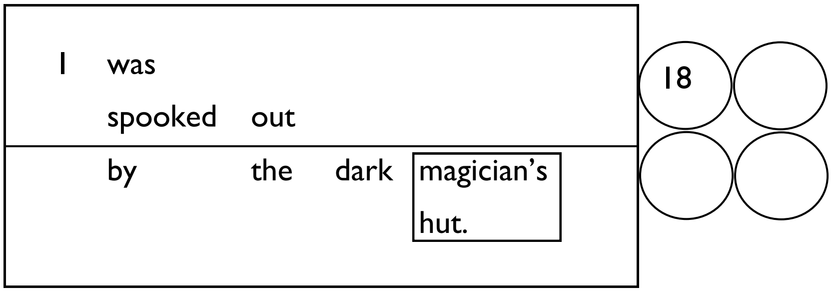
14. Please add the raisins by shaking them.

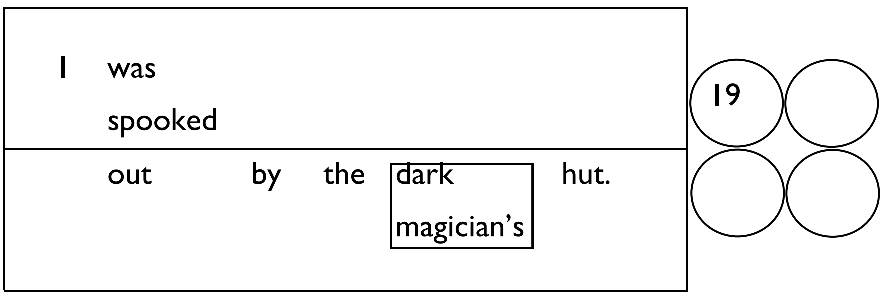
15. Please get into the raisins and shake yourself.

16. If you are going to shake yourself, please do it in the raisins.

(H) Twodee (5/10)

G.





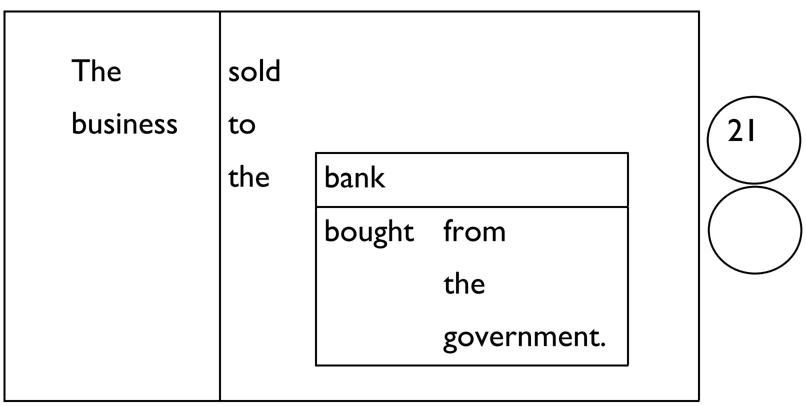
17. The hut of the dark magician spooked me out.

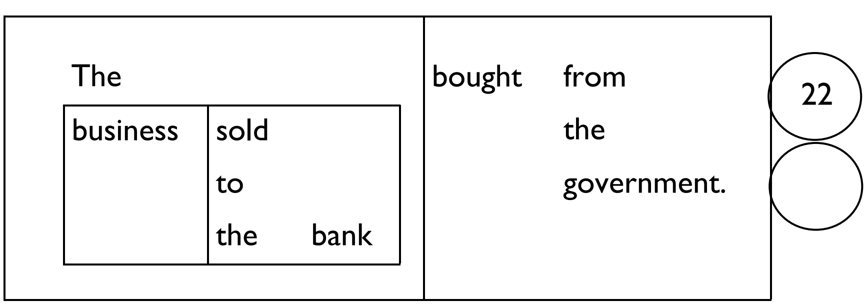
18. The dark hut of the magician spooked me out.

19. Out by the hut of the dark magician, something spooked me.

20. Out by the dark hut of the magician, something spooked me.

H.





21. The business sold things to the bank that was bought from the government.

22. The business that was sold to the bank bought things from the government.

(H) Twodee (6/10)

H2. Here is one correct answer (again, we've used boxes to indicate the most important phrases for you):

Zooey	left	her	husband	Death	Cab	for	Cutie	
Deschanel				frontman				
				Ben	Gibbard.			

And for good measure, here is a way of writing the editor's misinterpretation:

Zooey	left	her	husband	for	Cutie	frontman		
Deschanel		Death	Cab		Ben	Gibbard.		

(H) Twodee (7/10)

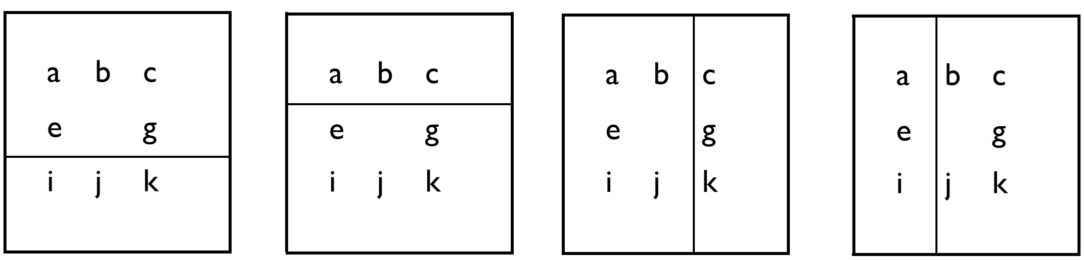
H3. In the original 8-word sentence, there are 7 places where two phrases are glued together horizontally. In each case, they could instead have been glued together vertically. Since there are 7 two-way choices, there are 2·2·2·2·2·2·2 = 128 ways to write the sentence.

Each of these 128 sequences of choices really does result in a *different* two-dimensional layout. Why? Any two different sequences must differ on at least one specific choice. For example, they might differ on whether "A computer" will be combined horizonally or vertically with "can count things." Regardless of the other choices, the first sequence must result in a layout where "A" is in the same row as "can" (but a different column), while the second sequence must result in a layout where "A" is in the same column as "can" (but a different row). So the two sequences cannot possibly give the same layout.

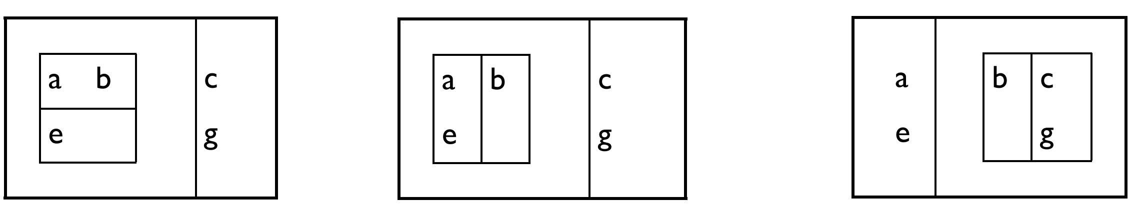
In general, a sentence with N+1 words is always built up by gluing N pairs of phrases together, so there are always 2^N ways to write it in Twodee -- provided that all of the N+1 words are different. (What if they're not? Can you construct a short sentence with fewer than 2^N layouts?)

H4. A. The correct answer is 6. <u>Incorrect answers that might get some credit include 8, 12, and 16</u>; these overlook the fact that some divisions give rise to the same spoken ordering.

Explanation: The top-level division must be one of the following.



Let's consider the first configuration. It doesn't matter how we divide up the bottom half (**ij | k** or **i | jk**), because the spoken order is **ijk** in either case. The top half can only be divided up vertically because the second row (**e [space] g**) is not a valid Twodee phrase (there is no way to build it with the space). The possible ways of dividing the top half are

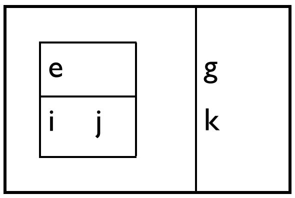


Spoken order: **abecgijk** Spoken order: **aebcgijk** Spoken order: **aebcgijk** (same)

which result in the spoken word orders shown underneath each drawing.

(H) Twodee (8/10)

Similarly, let's consider the second configuration. The top half has word order **abc** no matter how we divide it, and the bottom half again can only be divided vertically. In fact, because of the space, the only way to divide the bottom half is



Spoken order: **abceijgk**

So there are 3 distinct spoken orders for the first two configurations: **abecgijk**, **aebcgijk**, and **abceijgk**. The third and fourth configurations are symmetric to these (along the diagonal from the upper left to the lower right), and give another 3 spoken orders.

B. 14. Partial credit for 64.

C. 48. Partial credit for 770.

D. 274. Partial credit for 19,450.

Problems 4B-4D. encouraged you to *systematically* consider all ways that a large linguistic structure could have been built up from smaller ones.

Computer programs must do this in order to diagram, understand, or translate a sentence, even if it's a one-dimensional sentence. In effect, they consider all possible structures and use statistics to guess which one is right. But there are very many possible structures. The programs are clever enough that they manage to find the best structure without considering the structures one by one.

Similarly, you can figure out how many Twodee structures there are without counting them one by one. The Twodee problem lets you think about such algorithms in a pure form. You don't have to worry about the specific grammar rules of a particular language. And because you're only *counting* structures, you don't have to guess which one is right.

The key idea is that a large Twodee rectangle can be divided into two smaller Twodee phrases in multiple ways -- anywhere you can draw a horizontal or vertical line.

Consider one such division. If the first phrase has 7 possible spoken orders, and the second phrase has 5 possible spoken orders, then overall that division has $7 \cdot 5 = 35$ spoken orders. Multiplication is the trick that lets us avoid counting them individually.

(H) Twodee (9/10)

You have to add this 35 to the numbers you'd get from other divisions.

But where did those numbers 7 and 5 come from? From solving smaller versions of the problem (counting the number of orderings for the smaller rectangles). So the answer to problem d. builds on the answer to c., which builds on the answer to b.

By following this recipe instinctively by hand, it is possible to work out particular solutions to 4B-4D. But for generality, let's now write down the exact recipe for any grid with M rows and N columns. We will use the fact that there are no spaces and no repeated words.

First, let's solve the easier problem of counting the number of divisions (drawings with nested boxes). Let's write $f(M, N)$ for the number of divisions. Then

$$
\begin{aligned}
f(M, N) = \quad & 1 && \text{if } M=1, N=1 \text{ (no divisions)} \\
& + \sum_{i=1}^{M-1} f(i, N) \cdot f(M-i, N) && \text{if } M>1 \text{ (sum over horizontal divisions)} \\
& + \sum_{i=1}^{N-1} f(M, j) \cdot f(M, N-j) && \text{if } N>1 \text{ (sum over vertical divisions)}
\end{aligned}
$$

It is not too slow to build up the f values up through $f(4, 4)$, especially if you use the fact that $f(M, N) = f(N, M)$ by symmetry. Here for your interest is an even larger table of f values.

$f(1, 1) = 1$	$f(1, 2) = 1$	$f(1, 3) = 2$	$f(1, 4) = 5$	$f(1, 5) = 14$
$f(2, 1) = 1$	$f(2, 2) = 2$	$f(2, 3) = 8$	$f(2, 4) = 45$	$f(2, 5) = 318$
$f(3, 1) = 2$	$f(3, 2) = 8$	$f(3, 3) = 64$	$f(3, 4) = 770$	$f(3, 5) = 13{,}008$
$f(4, 1) = 5$	$f(4, 2) = 45$	$f(4, 3) = 770$	$f(4, 4) = 19{,}450$	$f(4, 5) = 729{,}148$
$f(5, 1) = 14$	$f(5, 2) = 318$	$f(5, 3) = 13{,}008$	$f(5, 4) = 729{,}148$	$f(5, 5) = 57{,}378{,}464$

We gave partial credit for answers from this table. However, while *f* accurately counts the number of divisions, it overcounts the number of spoken orderings. For example, $f(1, 3)=2$ because **abc** can be divided into **ab | c** or **a | bc**, and yet both of these divisions happen to give the spoken ordering **abc**.

To block the extra orderings, let's impose a restriction that if you divide a rectangle into 3 vertical stripes, for example, as we did with **abc**, then you have to divide it as **a | bc** rather than **ab | c**. Both give the same ordering. The general rule is that if you divide the rectangle vertically, you can't then immediately divide the *first* half vertically again, although you can still do so to the *second* half. We impose the same restriction on horizontal divisions.

(H) Twodee (10/10)

So let's define a new function $g(M, N, T)$ where T has one of the following values:

all: count all orderings.
noH: exclude orderings whose top-level division is horizontal.
noV: exclude orderings whose top-level version is vertical.

$$f(M, N, T) = 1 \qquad \text{if } M=1, N=1 \text{ (no divisions)}$$

$$+ \sum_{i=1}^{M-1} f(i, N, \text{noH})\cdot f(M-i, N, \text{all}) \qquad \text{if } M>1 \text{ and } T\neq\text{noH (sum over horizontal divisions)}$$

$$+ \sum_{j=1}^{N-1} f(M, j, \text{noV})\cdot f(M, N-j, \text{all}) \qquad \text{if } N>1 \text{ and } T\neq\text{noV (sum over vertical divisions)}$$

This gives us the following values. Note that as you might expect, $g(M, N, \text{all}) = g(M, N, \text{noH}) + g(M, N, \text{noV})$ except when $M=N=1$. Also, by symmetry, $g(M, N, \text{noH})=g(N, M, \text{noV})$.

g(1, 1, all) = 1	g(1, 2, all) = 1	g(1, 3, all) = 1	g(1, 4, all) = 1	g(1, 5, all) = 1
g(2, 1, all) = 1	g(2, 2, all) = 2	g(2, 3, all) = 4	g(2, 4, all) = 8	g(2, 5, all) = 16
g(3, 1, all) = 1	g(3, 2, all) = 4	g(3, 3, all) = 14	g(3, 4, all) = 48	g(3, 5, all) = 164
g(4, 1, all) = 1	g(4, 2, all) = 8	g(4, 3, all) = 48	g(4, 4, all) = 274	g(4, 5, all) = 1,548
g(5, 1, all) = 1	g(5, 2, all) = 16	g(5, 3, all) = 164	g(5, 4, all) = 1,548	g(5, 5, all) = 14,294

g(1, 1, noH) = 1	g(1, 2, noH) = 1	g(1, 3, noH) = 1	g(1, 4, noH) = 1	g(1, 5, noH) = 1
g(2, 1, noH) = 0	g(2, 2, noH) = 1	g(2, 3, noH) = 3	g(2, 4, noH) = 7	g(2, 5, noH) = 15
g(3, 1, noH) = 0	g(3, 2, noH) = 1	g(3, 3, noH) = 7	g(3, 4, noH) = 33	g(3, 5, noH) = 133
g(4, 1, noH) = 0	g(4, 2, noH) = 1	g(4, 3, noH) = 15	g(4, 4, noH) = 137	g(4, 5, noH) = 1,011
g(5, 1, noH) = 0	g(5, 2, noH) = 1	g(5, 3, noH) = 31	g(5, 4, noH) = 537	g(5, 5, noH) = 7,147

g(1, 1, noV) = 1	g(1,2,noV) = 0	g(1,3,noV) = 0	g(1,4,noV) = 0	g(1,5,noV) = 0
g(2, 1, noV) = 1	g(2,2,noV) = 1	g(2,3,noV) = 1	g(2,4,noV) = 1	g(2,5,noV) = 1
g(3, 1, noV) = 1	g(3,2,noV) = 3	g(3,3,noV) = 7	g(3,4,noV) = 15	g(3,5,noV) = 31
g(4, 1, noV) = 1	g(4,2,noV) = 7	g(4,3,noV) = 33	g(4,4,noV) = 137	g(4,5,noV) = 537
g(5, 1, noV) = 1	g(5,2,noV) = 15	g(5,3,noV) = 133	g(5,4,noV) = 1,011	g(5,5,noV) = 7,147

The answers to questions 4B-D. are g(3, 3, all) = 14, g(3, 4, all) = 48, and g(4, 4, all) = 274.