June 2007 LSAT

GAME #1: Numbered Ordering

Conceptualize the Game

This game requires that we order a set of five elements: 0, 1, 2, 3, and 4. This is a unique situation in that we are asked to order elements that are actually *numbers* instead of letters. This can cause some confusion if we're not careful. We must be sure not to confuse the *elements to be ordered* (0, 1, 2, 3, 4) with the *positions in which they will be placed* (1st, 2nd, 3rd, 4th, 5th). Given the fact that each code must be a "five-digit" code, we can start by creating a Number Line to represent the five positions. We'll put the elements to be ordered off to the side:

0
1
2
3
4

_____ _____ _____ _____ _____

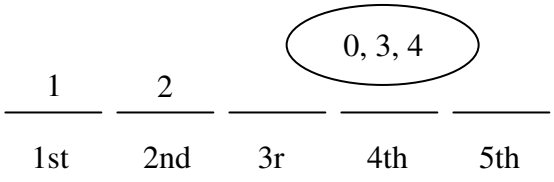
1st 2nd 3r 4th 5th

The second constraint gives us important information: Each digit occurs exactly once in any code. This means that any valid code will use each one of the five elements (0, 1, 2, 3, 4) once and only once (no repeats), and since we're dealing with a five-digit code, each slot above must be filled (no slots will be left empty). This may seem obvious for this particular game, but it's always important to "conceptualize" the game in this way before you begin. To help you conceptualize, always ask yourself the following questions: Will any of the elements be used more than once? Will any of the elements go unused? Will any of the slots be left unfilled? If you can get all of this straight before you begin, your task will be made much easier.

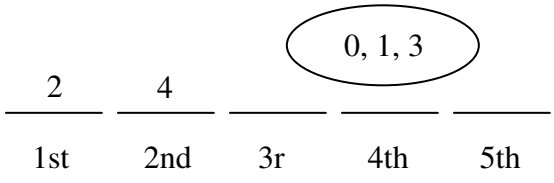
Setup

Let's see how we can use the constraints to make some inferences about the product codes. We know that the second digit has a value that is twice the value of the first digit. This constraint limits the possibilities for the 1st and 2nd slots. The only way that the second digit can be twice the first is if we have 1 and 2 or 2 and 4 in the first two slots. We'll put a "bubble" around the remaining three digits in each case to indicate that these digits will occupy the last three slots, though in an uncertain order. Here are the two possible frames:

Frame

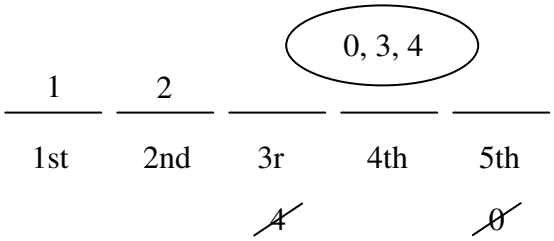


Frame

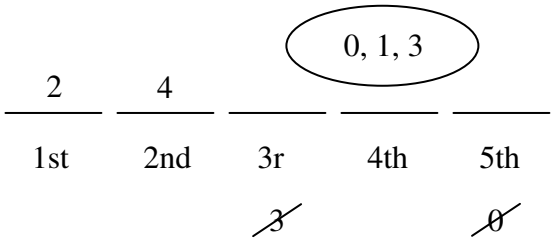


We also know that the value of the third digit must be less than the value of the last digit. In Frame #1, this means 4 can't go in the 3rd slot and 0 can't go in the 5th slot. In Frame #2, this means 3 can't go in the 3rd slot and 0 can't go in the fifth slot. We'll indicate this by putting some cross-outs underneath:

Frame



Frame



The Questions
At this point we're ready to apply our diagrams to the questions.

1. (A)
Question Type: Conditional
This question provides us with a piece of conditional ("If") information that pertains only to this question: the last digit is 1. What inferences can we make from this new information? Looking at our two frames above, we see that 1 can be last only in frame #2. In fact, if 1 is last in frame #2, then the product code would have to be: 2 4 0 3 1.

(A) Yes! We must have a 2 in the 1st slot. We can choose this and move on.

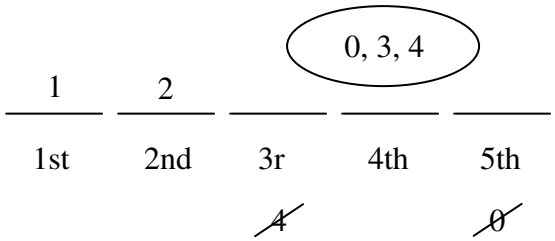
2. (C)
Question Type: Unconditional
This question does not provide us with any conditional ("If") rule in the question. It asks which of the answers "must be true." Thus, if we can prove an answer to be *untrue* in at least one case, that answer is wrong.

(A) The digit 1 does not *always* have to appear before the digit 2. In any frame #2 arrangement, the 1 will appear *after* the 2, not before it. Eliminate it.
(B) The digit 1 does not *always* have to appear before the digit 3. In frame #2, we could have 2 4 0 3 1, which puts the 1 *after* the 3, not before it. Eliminate it.
(C) Yes! This must be true. The 2 must come before the 3 in any arrangement.

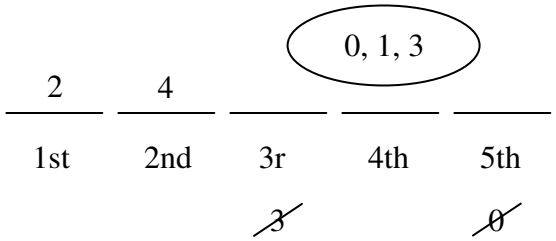
3. (C)
Question Type: Conditional
This question provides us with a piece of conditional ("If") information that pertains only to this question: the third digit is NOT 0. What inferences can we make from this information? In frame #1, if 0 is NOT in the 3rd slot it must go in the 4th slot (remember, 0 can't go 5th). We'd have: 1 2 3 0 4. We can use the same logic in frame #2. The 0 must go in slot 4, which means we'd have: 2 4 1 0 3. These are the only two possibilities. We're looking for an answer that "must be true."

(A) The second digit could be 2, but it need not be 2. Again, frame #2 would give us 2 4 1 0 3, which puts 2 in the first position. Eliminate it.
(B) The third digit could be 3, but it need not be 3. Frame #2 would give us 2 4 1 0 3, which puts 3 in the fifth position. Eliminate it.
(C) Yes! The 0 is in the 4th slot in both cases: 1 2 3 0 4 and 2 4 1 0 3. We can move on without checking (D) and (E).

Frame



Frame



4. (E)
Question Type: Unconditional
This question provides us with no conditional ("If") information. This is a tricky EXCEPT question. It's important that we anticipate the results before we begin: four of the answers *could* be valid in one or more cases, and one answer will *never* be valid. We're looking for the one answer that can *never* be valid.

(A) This could be true. 0 and 1 could be third and fourth. Frame #2: 2 4 0 1 3. Eliminate it.
(B) This could be true. 0 and 3 could be third and fourth. Frame #1: 1 2 0 3 4. There's another case in frame #2 that puts 0 and 3 in the third and fourth slots, but all we need is one case to prove it could be true. Eliminate it.
(C) This could be true. 1 and 0 could be third and fourth. Frame #2: 2 4 1 0 3. Eliminate it.
(D) This could be true. 3 and 0 could be third and fourth. Frame #1: 1 2 3 0 4. Eliminate it.
(E) This is impossible. If 3 and 4 are in the third and fourth slots (frame #1), that would put 0 in slot 5. We already know this can't be true.

5. (E)
Question Type: Unconditional
This question provides us with no conditional ("If") information. We're looking for an answer that "must be true" in all cases. Thus, if we can find at least one case that proves an answer *untrue*, we can eliminate that answer.

(A) There doesn't have to be exactly one digit between the 0 and 1. For example, frame #2 could give us: 2 4 0 1 3. This puts no digits between the 0 and 1. Eliminate it.
(B) There doesn't have to be exactly one digit between the 1 and 2. Any arrangement from frame #1 puts the 1 and the 2 adjacent to each other. Eliminate it.
(C) We can actually get *more* than two digits between the 1 and the 3. From frame #1 we could have: 1 2 0 4 3. This puts three digits between the 1 and 3. Eliminate it.
(D) We can actually get *more* than two digits between the 2 and the 3. From frame #2 we could have: 2 4 0 1 3. This puts three digits between the 1 and 3. Eliminate it.
(E) Yes! In no case can we get more than two digits separating the 2 and 4. This must be true.

NOTE: Before tackling the questions, you may have decided to list out all the possible product codes for this game. We didn't take this approach, but it is certainly possible (as it turns out, there are only six possible product codes that meet the constraints). This is a VERY unique game in this regard. Generally, it is NOT a good time investment to go down the road of listing out all the possibilities. In fact, most of the time, you'll get partway there and realize that there are more possibilities than you anticipated. While there are some very rare cases for which this will work, you should *always* be able to do the game *without* listing all the possibilities (as we did above).