1. find a number for x such that the remainder of 2x – 8 divided by 9 is zero: x = 4 + 9y, **x = 13** 🡪 **D**.
2. By the contrapositive, David can conclude that it did not snow that morning. However, just because it’s not cloudy in the afternoon does not mean that Hilary won’t swim at night; even if it’s not cloudy, Hilary can still swim (F->T is still true) (assuming otherwise would be the fallacy of concluding based on logical inverse). Thus the answer is **A**.
3. Working our way inside the logic expression:

There exists a magician AND (with the property):

There exist two distinct dragons that are owned by the magician AND

The magician does not own any dragons but these two dragons.

Thus, we know that **the magician owns two dragons** 🡪 **D**.

1. We need to capture two facts: there is a fastest dragon and no magician owns this dragon. These are captured in answer choice **C**. Choice A is incorrect because it only says that some magician doesn’t own the fastest dragon (but not that NO magician owns the fastest dragon). Choice B is incorrect because it says that among all magicians, one of the dragons they own is not the fastest. Choice D is incorrect because it says that given the fastest dragon, it is not owned by every single magician (which allows it to be owned by some magicians).
2. You can solve this by creating a propositional formula, (e.g. the information in a person’s statement is truth if and only if “<->” the person is an oracle), and then make a truth table for that formula. Alternatively, we can just reason through these statements:

* Suppose X is an oracle. Then both X and Y must be oracles based on X’s statement; but this implies that Y is an oracle. This creates a contradiction because Y says that X must be a necromancer (contradicting our assumption that X is an oracle).
* Thus, X must be a necromancer. Suppose that Y is an oracle; then everything is logically sound; so **X is a necromancer and Y is an oracle** 🡪 **D**.
  + Note that Y CANNOT be a necromancer because that would mean that Y’s statement “X is a necromancer” would violate the rule that necromancers ALWAYS lie (because X is a necromancer, so Y would be telling the truth instead of lying).

1. First prime factor the number: . Let x be a factor of 7920, then x can be a combination of . So as a general trick, if we take the exponent of each prime factor {4,2,1,1}, add one to the value of each exponent {5,3,2,2} (to account for raising the prime factor to the 0th power), and multiply these values together, we get the total number of positive, integral factors: 5\*3\*2\*2 = **60** 🡪**D**.
2. **🡪 C**
3. Numbers with a units digit of 7 will repeat units digits in a pattern:

So we can use remainder\_of(exponent/4) to determine what the units digit will be: 23/4 yields a remainder of 3, so the units digit will be  **🡪 B.**

1. **Choice C is** **FALSE**: This is only true if both a and b are relatively prime; however, answer choice C only says that they’re arbitrary integers, so consider n = 60 which is divisible by 20 and 30, but is clearly not divisible by 20 \* 30 = 600.

Choice A is true: Trivially, if two numbers are relatively prime, then they share no integral factors, so they can’t share a prime factor. Conversely, if two integers do not share any prime factors then they cannot share any integral factor (because every integer can be written as a product of primes by the Fundamental Theorem of Arithmetic).

Choice B is true: Let *x* be the factor that is greater, then for some y by definition of factor. But , so .

Choice D is true: This is the Fundamental Theorem of Arithmetic – all integers greater than 1 have a prime factorization.

1. The correct answer is 48. By the Fundamental theorem of Arithmetic, we know that every positive number has a prime factorization. Furthermore, we know that for each of the exponents on each prime factor, if we add one to the value of the exponent and multiply all those values together, we get the total number of factors for that number: has factors. Thus, when we’re looking for a number with only 10 factors, we need a number of the form or . Substituting in the smallest prime numbers for p1/p2, we get that is the smallest possible number. **E**.
2. Principle of Exclusion-Inclusion for Sets: 🡪 **A**.
3. Claim I is ***not*** always true. Because we’re dealing with element of, rather than subset of, there isn’t an always-true transitive relation. Consider the following counter example: A = the empty set: ,

B = the set containing the empty set: , and C = the set containing the set containing the empty set: . A is NOT contained in C because the only element in C is the set containing the empty set.

Claim II is ***not*** always true. If A = the empty set: and B = the set of real numbers: , then A – B = and .

Claim III is ***always*** true. To show A = B, we need to show that A is a subset of B and that B is a subset of A: by definition of power-sets, A is an element of p(A) and therefore p(B) because p(A) = p(B). Thus A is a subset of B because A is an element of B’s power-set (which is the set of all subsets of B).

Similarly, B is an element of p(A) = p(B) so therefore B is a subset of A. Thus, A = B 🡪 **E**.

1. The Cartesian Product, , is defined as the set . Thus the correct answer is B by applying the definition; the elements of set B are sets themselves, so the b-ordinate in the pair is just a set-element in set B. **B**
2. Function II grows the slowest because and grows slower than. clearly grows faster than x because ln(x) grows faster than the constant 1. By Sterling’s approximation, , so grows faster than . Using Sterling’s approximation again, we know that grows slower than because . So the ordering from slowest to fastest is II, I, III, IV 🡪 **C**.
3. We could find a closed-form formula for the recursion, but since the computation is low enough, we can just unroll the recursion: f(1) = 1, f(2) = 1, f(3) = 5 – 2 = 3, f(4) = 15 – 2 = 13, f(5) = 65 – 6 = 59 🡪 **B**.
4. Statement I is true – the function f(x) = 0 is both even and odd.

Statement II is true – let f(x) and g(x) be two injective functions. f(g(x)) must be injective because g(x) corresponds to a unique value y (because g is injective). Therefore, f(g(x)) maps to a unique value z because f(y) maps to a unique z because f is injective.

Statement III is true – let f(x) and g(x) be two surjective functions. Then f(g(x)) is surjective because g(x) covers every value in f’s domain by definition of surjective and f is surjective, so f(g(x)) covers every value in f’s range 🡪 **D**.

1. Each digit can be considered an independent event and only care about the last two digits/events. Pr[least significant digit = 0] = ½ and Pr[2nd least significant digit = 0] = ½. The intersection of independent events multiples so, Pr[two least significant digits = 0] = ½ \* ½ = **¼** 🡪 **B**.
2. Using the inclusion-exclusion principle: 🡪 **D**.
3. First, find the number of possible regroupings. Consider selecting one pair at a time. There are ways to form the first pair. Out of the remaining eight individuals, there are possible second pairs. Similarly there would remain possible third pairs, possible fourth pairs, and possible fifth pairs. So, there are possible ways to form a first, second, third, fourth, and fifth pair. However, the ordering of these pairs doesn’t matter. There are 5! possible ways to order the five pairs, so there are total possible ways to form the pairs. Only one of these pairings results in all five twin pairs remaining intact, and every pairing is equally likely, so the probability that all individuals get paired with their twin is 🡪 **C**.
4. If Michela buys two Fury Stones of each type, she won’t have enough to make her earring; this would be 16 stones. However, if she buys one additional stone, the Pigeonhole Principle guarantees that she’ll have at least three Fury Stones of one type; so she needs to buy 17 stones. Thus the answer is **E**.
5. Answer choices A, B, D all upper bound f(x) as the limit of x approaches infinity. However, no matter what constants “c” and “k” you choose for , the limit of will outgrow because the square root of x grows faster than the log of x. **C**
6. The outer for-loop adds a (n) complexity to the runtime (because it will execute “n” times). The inner while-loop contributes a complexity to the runtime for each execution of the for-loop. Thus the to overall big- runtime is , where the log-base 2 constant factor is thrown away because of big- notation. **D**
7. Doing some row reduction by adding 2 times the first row to the third row, we get: A = which is an upper triangular matrix, so the determinant is just the product along the main diagonal = 7 \* 13 = **91 🡪 A**.
8. Choices A and B are false because the kernel has a dimension greater than zero (i.e. the linear map has more zeroes than just the vector 0), thus T cannot be injective; moreover, since T is a linear operator, since T is not injective, it cannot be invertible (since an invertible linear operator is injective and surjective). Choice D is false because the dimension of vector space V is independent of how T behaves on V. Choice C is correct by the Rank-Nullity Theorem. **C**.
9. Because the matrix of T is defined with respect to the standard basis, we can just multiply the matrix of T with the input vector (2, -1, 1) to get the value of T(2, -1, 1): A(2, -1, 1) = (4-4+1, 10-6+1, 6-2) = **(1, 5, 4) 🡪 C**.
10. Answer Choice **A** is correct. C must be a subset of A: Let x be an arbitrary element of , then because . Because, x is also an element of , which implies that . Thus , so C is a subset of A. The other choices can be ruled out with simple counter examples.
11. This proof is valid by reduction; as it stands, Grant’s proof reduces VeniBox lifting to lifting an anvil. In other words, he argues that in order to lift a VeniBox, he must be able to lift an anvil because any VeniBox can contain an anvil, so he must therefore be able to lift an anvil in order to lift a VeniBox. Using the contrapositive on this reduction, Grant can conclude that because he can’t lift an anvil, he can’t lift a VeniBox. **C**
12. This algorithm is Prim’s MST Algorithm. We loop until all the nodes in the graph are connected (thus spanning tree) and we greedily take only the minimum weighted edge at each step (thus the spanning tree is a minimum spanning tree). **B**
13. This is the same as finding the number of ways we can group the nodes into pairs (number of handshakes that “n” people can make): 15 choose 2 (combinations) = 15 \* 14/2 = 85 🡪 **E**.
14. The correct answer is 8. If each city had seven roads to seven other towns, that would mean we would have 7 \* 11 = 77 complete roads; but that double counts each road, so we need to divide it by two yield 77/2 = 38.5 complete roads – but there’s no such thing as 0.5 complete roads, so we can’t have exactly seven roads per town. However, it is possible to connect each town to eight other towns using 44 roads. **B**